

На правах рукописи



**Лапутенко Андрей Владимирович**

**АНАЛИЗ И СИНТЕЗ ЛОГИЧЕСКИХ СХЕМ ДЛЯ ПРОВЕРКИ  
ФУНКЦИОНАЛЬНЫХ И НЕФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ  
ДЛЯ КОМПОНЕНТОВ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ**

05.13.01 – Системный анализ, управление и обработка информации  
(в отраслях информатики, вычислительной техники и автоматизации)

Автореферат  
диссертации на соискание ученой степени  
кандидата технических наук

Томск – 2021

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Национальный исследовательский Томский государственный университет».

**Научный руководитель:** доктор технических наук, профессор  
**Евтушенко Нина Владимировна**

**Официальные оппоненты:**

**Матророва Анжела Юрьевна**, доктор технических наук, профессор, федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет», кафедра компьютерной безопасности, профессор

**Бурдонов Игорь Борисович**, доктор физико-математических наук, старший научный сотрудник, Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук, отдел «Технологий программирования», ведущий научный сотрудник

**Бразовский Константин Станиславович**, доктор технических наук, федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский политехнический университет», Исследовательская школа химических и биомедицинских технологий, профессор

Защита состоится 02 июня 2021 г. в 10 час. 30 мин. на заседании диссертационного совета «НИ ТГУ.05.01», созданного на базе Института прикладной математики и компьютерных наук федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Томский государственный университет», по адресу: 634050, г. Томск, пр. Ленина, 36 (учебный корпус № 2 ТГУ, аудитория 212Б).

С диссертацией можно ознакомиться в Научной библиотеке и на официальном сайте федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Томский государственный университет» [www.tsu.ru](http://www.tsu.ru).

Материалы по защите диссертации размещены на официальном сайте ТГУ: <https://dissertations.tsu.ru/PublicApplications/Details/e3fa1747-3bc9-4849-8c4f-8c5dcaab68f8f>

Автореферат разослан «\_\_» апреля 2021 г.

Ученый секретарь  
диссертационного совета

Нежелская  
Людмила Алексеевна

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы исследования.** Логические схемы являются одной из первых формальных моделей, которые были использованы для анализа и синтеза цифровых систем и устройств, и логические схемы активно исследуются с начала 40-х годов XX века. В последнее время по ряду причин интерес к исследованиям логических схем получил «новый виток». Одной из таких причин является возможность эффективного использования логических схем для проверки функциональных и нефункциональных требований для программных и аппаратных компонентов киберфизических и телекоммуникационных систем. В частности, логические схемы можно использовать в качестве формальных моделей для программных и аппаратных компонентов с большим числом состояний при построении тестов с гарантированной полнотой. Еще одним направлением, получившим широкое распространение в последнее время, является использование логических схем для представления различных моделей машинного обучения, в частности, для оценки качества телекоммуникационных сервисов, т.е. для проверки нефункциональных требований. Данная задача является актуальной для приложений, выполняющихся на устройствах с ограниченными вычислительными возможностями, использующихся в интернете вещей (Internet of Things, IoT). Суть технологии IoT заключается в объединении большого числа малогабаритных устройств, обменивающихся друг с другом в том числе и конфиденциальными данными (различные датчики, «умные» устройства и т.д.). Соответственно, при решении задач безопасности важным шагом является определение характеристик доверия (англ. trust) к приложению, выполняющемуся на подобном устройстве, для чего часто используют различные модели машинного обучения. Поскольку подобные устройства снабжаются маломощными процессорами, небольшим количеством памяти и аккумуляторами небольшой емкости, это сильно затрудняет запуск на них дополнительных приложений, особенно таких ресурсоемких как программные реализации алгоритмов машинного обучения. В настоящее время для эффективного представления алгоритмов машинного обучения, в том числе нейронных сетей, предлагается использовать логические схемы, точность предсказания которых оказывается близкой, а в некоторых случаях равной точности нейронной сети, по которой строится логическая схема.

**Целью работы** является разработка математических, программных и аппаратных средств для анализа и синтеза логических схем для проверки функциональных и нефункциональных требований для компонентов телекоммуникационных систем.

Для достижения поставленной цели, необходимо решить следующие задачи.

1. Разработать алгоритмы синтеза тестов для проверки функциональных требований на различных уровнях абстракции компонентов телекоммуникационных систем, в том числе для временных конечных автоматов; сравнить качество и сложность построения тестов. Экспериментально оценить качество построенных тестов.

2. Разработать методику аппаратной реализации моделей машинного обучения с использованием логических схем для устройств с ограниченными вычислительными возможностями при оценке характеристик доверия для компонентов интернета вещей.

### **Научная новизна.**

1. Предложены алгоритмы синтеза тестов для дискретных систем на различных уровнях абстракции с использованием логических схем, в том числе для временных автоматов. Более конкретно, предложена методика построения тестов для логических схем, объединяющая конечно-автоматные методы и методы систем логического синтеза и верификации при построении различающих последовательностей. Проведено экспериментальное сравнение качества тестов, построенных по логической схеме и модели (временного) конечного автомата, которое показало, что тесты, построенные по логической схеме, являются более короткими и имеют удовлетворительную полноту обнаружения неисправностей.

2. Исследована эффективность построения мутационных тестов по модели логической схемы для проверки функциональных свойств (временных) дискретных систем.

3. Предложена методика представления моделей машинного обучения в виде логических схем для последующей эффективной аппаратной реализации на ПЛИС. Предложенная методика может быть использована для интеграции в устройства с ограниченными вычислительными возможностями некоторого модуля, проверяющего нефункциональные требования, включая оценивание характеристик доверия для приложений, работающих в интернете вещей.

**Теоретическая и практическая ценность.** В работе предложены алгоритмы построения проверяющих тестов на основе логических схем для проверки функциональных требований к системам, поведение которых задается (временными) конечными автоматами. Алгоритмы опробованы при синтезе тестов для ряда микроконтроллерных реализаций, в частности, при тестировании реализации генератора импульсов запуска CCD-камеры. Предложена методика для представления моделей машинного обучения в виде логических схем, в том числе для последующей эффективной аппаратной реализации в виде ПЛИС; аппаратная реализация может быть использована в устройствах с ограниченными вычислительными

возможностями для проверки нефункциональных требований к приложениям, которые выполняются на устройстве, в частности, для компонентов интернета вещей. В работе такое расширение функций устройства используется для оценивания характеристик доверия к приложению. Эксперименты, проведенные с программной реализацией таких приложений, показали эффективность предложенной методики. Результаты диссертационной работы используются в АО «НПФ «Микран» (Томск) для повышения качества тестов для штатных цифровых компонентов за счет генерации дополнительных тестов, в АО «НВП «Топаз» (Томск) при синтезе тестов для проверки функциональных и нефункциональных требований к управляющим системам, реализованным на микроконтроллерах, а также в учебном процессе на кафедре информационных технологий в исследовании дискретных структур Национального исследовательского Томского государственного университета.

**Методы исследования.** Для решения задач синтеза проверяющих тестов используется аппарат дискретной математики, в частности, аппарат классической теории автоматов, включая теорию логических схем, математической логики, а также методов логического синтеза и верификации. Для оценивания выполнения нефункциональных требований используются модели и методы машинного обучения. Оценка качества предложенных алгоритмов проводится с помощью компьютерных экспериментов, в том числе с использованием ПЛИС Altera.

**Основные положения, выносимые на защиту:**

1. Методика построения тестов с гарантированной полнотой для логических схем на основе конечно-автоматных методов и верификаторов, в том числе относительно неисправностей, возникающих при замене вентиля схемы.

2. Алгоритм синтеза тестов на основе логических схем, доставляющий тесты с гарантированной полнотой относительно неисправностей (временного) конечного автомата.

3. Методика синтеза логической схемы по модели машинного обучения для проверки характеристик доверия к приложениям, выполняющимся на устройствах с ограниченными вычислительными возможностями; логическая схема далее может быть использована для аппаратной реализации модели машинного обучения.

4. Экспериментальные результаты, которые подтвердили эффективность предложенных в диссертации алгоритмов.

**Достоверность результатов.** Все положения, представленные в диссертации, получены с использованием аппарата дискретной математики. Эффективность предложенных алгоритмов подтверждается компьютерными экспериментами, в частности, по оценке полноты тестов, построенных по мутациям логической схемы относительно неисправностей временного

конечного автомата. Результаты согласуются с известными результатами для константных и мостиковых неисправностей логических схем и классических конечных автоматов. Эффективность предсказательных способностей логической схемы при проверке нефункциональных требований подтверждается результатами компьютерных экспериментов с моделями машинного обучения, для которых возможна симуляция модели на всех входных наборах.

**Публикации.** По результатам проведенных исследований опубликовано 11 работ, в том числе 4 статьи в журналах, включенных в Перечень российских рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора наук, на соискание ученой степени кандидата наук (из них 2 статьи в российском научном журнале, переводная версия которого входит в Web of Science, 1 статья в российском научном журнале, входящем в Web of Science), 5 статей в сборниках материалов научных конференций, представленных в изданиях, входящих в Web of Science и / или Scopus, 1 статья в прочем научном журнале, 1 публикация в сборнике всероссийской научной конференции с международным участием.

**Личный вклад в положения, выносимые на защиту.**

Постановка задач, решаемых в диссертации, осуществлены соискателем совместно с научным руководителем, доктором технических наук, профессором Н. В. Евтушенко. В работах по теме диссертации диссертантом предложены ключевые научные идеи, проведены эксперименты, в опубликованных статьях написаны части, в которых изложены выносимые на защиту результаты.

**Апробация работы.**

Основные положения и результаты были представлены на следующих конференциях и семинарах: Российская конференция с международным участием «Новые информационные технологии в исследовании сложных структур» (г. Екатеринбург, Россия, 2016; пос. Катунь, Россия, 2018), Международная конференция молодых специалистов по микро/нанотехнологиям и электронным приборам EDM (Алтайский край, Россия, 2017, 2018, 2019), научно-технические семинары отдела по технологиям программирования ИСП РАН (Москва, Россия, 2018, 2020), Международная конференция по программным технологиям ICISOFT (Порту, Португалия, 2018), научно-технический семинар лаборатории логического проектирования ОИПИ НАН Беларуси (Минск, Республика Беларусь, 2020).

**Структура и объем работы.** Диссертация состоит из введения, 4 глав, заключения, списка литературы, включающего 122 наименования, и 3 приложений. Диссертация содержит 31 рисунок и 13 таблиц. Общий объем диссертации составляет 115 страниц.

## СОДЕРЖАНИЕ РАБОТЫ

Во **введении** приводится общая характеристика работы, обосновывается актуальность исследований, формулируются цель и основные задачи диссертации, кратко излагаются полученные результаты, научная новизна работы, практическая ценность и выносимые на защиту положения. Кроме того, во введении приводится краткий обзор результатов по теме диссертации.

В **первой главе** вводятся основные определения и обозначения.

Конечные системы переходов, такие как конечные автоматы, активно используются при решении задач анализа и синтеза для различных систем, которые переходят из состояния в состояние под действием входных воздействий, производя при этом выходные реакции.

Формально под *конечным автоматом* или просто *автоматом* будем понимать четверку  $\mathbf{S} = (S, I, O, T_S)$ , где  $S$  – непустое конечное множество *состояний*,  $I$  – непустое конечное множество входных символов, называемое *входным алфавитом*,  $O$  – непустое конечное множество выходных символов, называемое *выходным алфавитом*,  $T_S \in S \times I \times O \times S$  – множество *переходов*. Четверка  $(s, i, o, s') \in T_S$  называется *переходом* в автомате  $S$  и описывает ситуацию, когда на автомат, находящийся в состоянии  $s$ , поступает входной символ  $i$ , под действием которого автомат переходит в состояние  $s'$  с выдачей выходного символа  $o$ . Автомат  $S$  с выделенным начальным состоянием  $s_0$  называется *инициальным* и обычно описывает поведение систем, в которых существует надежный специальный сигнал СБРОС (*reset*), по которому система гарантированно переходит из любого состояния в начальное состояние. Автомат называется *детерминированным*, если из каждого состояния  $s$  существует только один переход для каждого входного воздействия, определенного в данном состоянии. Детерминированные автоматы активно исследовались в XX-м веке, в частности, в области построения тестов с гарантированной полнотой для телекоммуникационных протоколов, при разработке автоматизированных и автоматических методов синтеза управляющих систем, трансформирующих последовательности в одном (входном) алфавите в последовательности в другом (выходном) алфавите, и в этой области получен ряд основополагающих результатов. В большинстве работ рассматриваются полностью определенные автоматы, когда для каждой пары  $(s, i) \in S \times I$  существует пара  $(o, s') \in O \times S$ , такая, что  $(s, i, o, s') \in T_S$ ; иначе автомат называется *частичным*. Чтобы избежать использования частичных автоматов, в ряде случаев частичный автомат различными способами, соответствующими рассматриваемой задаче, доопределялся до полностью определенного, и в нашей работе мы в

большинстве случаев рассматриваем компоненты, поведение которых определено на каждой входной последовательности.

Под *временным автоматом*  $S$  понимается детерминированный автомат, в который добавлены входные и выходные таймауты. Таким образом, временной автомат есть семерка  $(S, I, O, \lambda_S, s_0, \Delta_S, \sigma_S)$ , где  $S, I$  и  $O$  – конечные непустые множества состояний, входных и выходных символов соответственно,  $s_0$  – начальное состояние,  $\lambda_S \subseteq S \times I \times O \times S \times Z$  – отношение переходов,  $\Delta_S: S \rightarrow S \times Z$  – функция таймаутов и  $\sigma_S: S \times I \times O \times S \rightarrow Z$  – функция выходных задержек (выходных таймаутов). Для  $\lambda_S$  через  $Z$  обозначено множество входных задержек (входных таймаутов), где  $Z$  есть множество натуральных чисел в объединении с числом 0. Если время ожидания в некотором состоянии превышает (входной) таймаут для этого состояния, то автомат может спонтанно перейти в другое состояние. В данной работе рассматриваются временные автоматы, у которых в каждом состоянии задержка выхода каждого перехода меньше величины таймаута. *Временным входным символом* называется пара  $(i, t)$ , где  $i$  – символ входного алфавита,  $t$  – (вещественное) время поступления входного символа после перехода автомата в текущее состояние. *Временным выходным символом* называется пара  $(o, d)$ , где  $o$  – символ выходного алфавита,  $d$  – выходная задержка. Последовательность временных входных символов  $(i_1, t_1) (i_2, t_2) \dots (i_n, t_n)$  называется *временной входной последовательностью*; последовательность временных выходных символов  $(o_1, d_1) (o_2, d_2) \dots (o_n, d_n)$  называется *временной выходной последовательностью*. Аналогично конечным автоматам, временная выходная последовательность, соответствующая временной входной последовательности  $\alpha$ , поступившей на автомат в состоянии  $s$ , называется (*выходной*) *реакцией* автомата в состоянии  $s$  на последовательность  $\alpha$ .

В ряде случаев поведение временного автомата можно описать специальным классическим автоматом, который называется *конечно-автоматной абстракцией автомата с таймаутами (временного автомата)*. В этом случае вводится специальный входной и выходной символ  $\Gamma$ , который соответствует единице времени, и все переходы в исходном временном автомате соответственно «растягиваются». Классический конечный автомат для временного автомата строится в два этапа. На первом шаге строится временной автомат с таймаутами, но без задержек выходов: для каждого выходного символа  $o_i$ , выдаваемого с некоторой задержкой  $\theta$ , в выходной алфавит вводится новый выходной символ  $\langle o_i, \theta \rangle$ . На втором шаге выполняется построение классического конечного автомата по заданному временному автомату с таймаутами: каждый переход  $s \xrightarrow{T} s'$  заменяется цепочкой переходов



$s \xrightarrow{I/I} s_1 \xrightarrow{I/I} s_2 \xrightarrow{I/I} \dots \xrightarrow{I/I} s_{T-1} \xrightarrow{I/I} s'$ , где каждое новое промежуточное состояние является копией состояния  $s$  по всем входо-выходным парам, кроме I/I. Конечный автомат, построенный по описанным выше правилам, является конечным автоматом специального вида и имеет ряд особенностей: во-первых, мы выделяем в этом автомате состояния, в которых автомат не может находиться дольше одного такта времени; во-вторых, входной и выходной алфавиты пересекаются, так как содержат один и тот же выделенный символ I. Этот символ обозначает отсутствие подачи какого-либо входного символа на автомат в течение одного такта времени, а также отсутствие какой-либо реакции автомата. Последнее не вызывает никаких проблем, так как эти символы можно снабдить различными индексами для обозначения входного и выходного символов.

Два состояния одного автомата (двух автоматов) называются *эквивалентными*, если множества реакций автомата(-ов) на все возможные входные последовательности совпадают для этих состояний. Два инициальных детерминированных полностью определенных автомата называются эквивалентными, если эквивалентны их начальные состояния. Если состояния не являются эквивалентными, то они называются *различимыми*. *Различающей последовательностью* называется входная последовательность, на которую автомат(-ы) производит(-ят) различные реакции в этих состояниях.

При синтезе тестов для конечного автомата вводится понятие модели неисправности, состоящей из трех элементов: автомата-спецификации, задающего эталонное поведение системы, области неисправности, задающей множество неисправных автоматов, и отношения конформности (эквивалентности). Под *тестовой последовательностью* понимают последовательность входных символов автомата. Множество тестовых последовательностей называется *тестом*. Тест называется *полным*, если он позволяет обнаружить каждый автомат из области неисправности, который неконформен (неэквивалентен) автомату-спецификации.

Одним из недостатков модели конечного автомата при решении задач анализа и синтеза для компонентов киберфизических и телекоммуникационных систем является его размерность. Современные программы достаточно быстро решают задачи синтеза и анализа автоматов, размеры которых не превосходят 1000 состояний и 10 входных и выходных символов. Для автоматов больших размерностей активно используется представление автомата в виде логической схемы. Логическая схема состоит из логических элементов, где под (логическим) *элементом* понимается  $(m, 1)$ -полюсник, который имеет  $m$  входных полюсов (входов), один выходной полюс (выход) и реализует некоторую логическую функцию. В данной работе элемент может быть комбинационным (*вентилем*), т.е. реализовывать

некоторую элементарную булеву функцию, либо элементом задержки, который имеет два состояния 0 и 1, два входных и два выходных символа. Такой элемент задержки часто называется *D-триггером* (обозначение: *D* или *d*). Следующим состоянием элемента задержки (триггера) является значение текущего входного символа, выходным символом является значение текущего состояния. Под *логической схемой* понимается пара  $\langle E, R \rangle$ , где  $E$  – совокупность элементов, имена которых попарно различны, и  $R$  – отношение эквивалентности, заданное на множестве полюсов элементов из  $E$ . Блоки разбиения  $R$ , содержащие выходной полюс, также могут быть объявлены выходными полюсами схемы. В данной работе рассматриваются схемы с синхронным функционированием, т.е. такие, в которых следующее состояние и выходной вектор вычисляются за один такт. *Комбинационная схема* состоит из взаимосвязанных логических элементов, не имеет элементов памяти и обратных связей и реализует в общем случае систему булевых функций. *Последовательностные схемы* (схемы с памятью) включают как комбинационную часть (комбинационную схему), так и элементы задержки (триггеры), и выходная реакция схемы зависит не только от текущего входного вектора, но и от предыдущих входных последовательностей. Пример последовательностной схемы изображен на рисунке 1. Данная логическая схема имеет один вход, два элемента задержки и один выход.

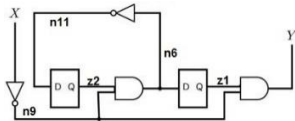


Рисунок 1 – Пример последовательностной схемы

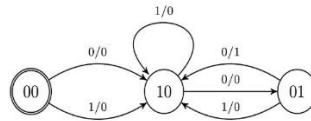


Рисунок 2 – Конечный автомат, описывающий поведение схемы на рисунке 1

Конечный автомат, соответствующий логической схеме, можно построить путем моделирования выходных реакций схемы во всех состояниях в ответ на входные воздействия. Множество состояний конечного автомата состоит из булевых векторов (при необходимости хешированных целыми числами), представляющих комбинации значений элементов задержки, достижимые из начальной комбинации (начального состояния). Соответственно, множество входных символов автомата содержит булевы векторы, которые могут быть поданы на входы схемы. Например, конечный автомат, описывающий поведение логической схемы, изображенной на рисунке 1, представлен на рисунке 2. Поведение логической схемы можно описать с использованием HDL (Hardware Description Language) языков, таких как языки Verilog и VHDL, и форматов описания цифровых схем, таких

как BLIF (Berkeley Logic Interchange Format) и bench (ISCAS'89). Для описания детерминированного автомата логической схемой нужно закодировать состояния, входные и выходные символы булевыми векторами подходящей длины и соответственно вычислить функции переходов и выходов автомата в виде системы булевых функций. В частности, использование представления логических схем в виде BDD уже дало возможность рассматривать дискретные системы с тысячами состояний, входных и выходных символов. Однако число элементов задержки обычно ограничивалось требованиями решаемой задачи, а входных и выходных портов было не более 50. Появление методов и алгоритмов решения задач анализа и синтеза для логических схем на основе решения проблемы выполнимости (SAT проблемы) предоставило возможность рассматривать схемы с большим числом элементов задержки, входных и выходных портов.

При синтезе тестов для логических схем, распространен мутационный подход, согласно которому вводится модель неисправности, содержащая множество схем с ошибочным поведением, так называемых «мутантов» спецификации. Для каждого мутанта строится различающая последовательность, позволяющая обнаружить мутанта. Полный тест состоит из множества входных последовательностей схемы-спецификации, позволяющих отличить всех рассматриваемых мутантов от схемы-спецификации. Зачастую множество мутантов формируется путем явного перечисления неисправных схем, и в диссертации рассматривается именно такой подход. Рассматриваются 3 типа неисправностей в логических схемах: одиночная константная неисправность, возникающая при установлении на одной линии схемы постоянного значения логического нуля или единицы; одиночная мостиковая неисправность (перемычка), возникающая при ошибочном появлении соединения между входами и выходами различных вентилях; одиночная трудно обнаружимая неисправность, возникающая при замене одного из вентилях схемы на близкий по поведению. Последний тип неисправностей стал распространен с развитием технологии ПЛИС, в которой описание схем дается на языках высокого уровня (Verilog, VHDL), и разработчик может легко допустить ошибку при описании одного логического элемента.

В разделе 6 главы 1 кратко описаны различные модели машинного обучения, применяемые для решения различных задач классификации, такие как искусственные нейронные сети (Artificial Neural Network, ANN), системы опорных векторов (Support Vector Machine, SVM). Модель машинного обучения принимает на вход *стимулы* и *реакции*. Ставится задача обучить модель сопоставлять реакции стимулам по имеющимся изначально ограниченному набору стимулов и реакций. Различными исследователями предложены алгоритмы для обучения систем опорных векторов, нейронных сетей и других моделей с высокой точностью предсказания.

Во **второй** главе исследуется задача синтеза тестов для проверки функциональных требований к цифровым системам на различных уровнях абстракции, а именно на уровне логических схем и конечных автоматов. Построенные тесты используются для проверки функциональных требований для различных систем, в том числе систем, реализованных на ПЛИС. Преимуществом модели конечного автомата является тот факт, что тесты строятся по ней без явного перечисления неисправностей, только по автомату-спецификации. Недостатком является размерность модели, не позволяющая строить тесты для компонентов с числом состояний больше 1000. Достоинством модели логической схемы является ее масштабируемость. Например, для системы с 1000 состояний достаточно использовать логическую схему с числом элементов задержки не больше 10. Результаты, представленные в данной главе, опубликованы в работах [1, 4, 5, 7].

В связи с развитием различных языков и форматов для описания логических схем и появлением новых технологий реализации схем (например, ПЛИС), множество неисправностей, рассматриваемых в логических схемах, было значительно расширено, поскольку константные неисправности были, в первую очередь, адекватными только для релейно-контактных схем. Тесты, строящиеся по моделям высокого уровня, таким как, например, расширенные автоматы, обладают низкой полнотой относительно ошибок в реализациях компонентов цифровых устройств. Согласно исследованиям, такие тесты оставляют необнаруженными большое число функциональных ошибок в программном обеспечении. Проверка качества тестов, построенных конечно-автоматными методами относительно неисправностей в логических схемах, проводилась в основном относительно (одиночных) константных неисправностей. Нам не известны подобные результаты относительно трудно обнаружимых неисправностей. Поэтому в настоящей работе мы исследуем качество тестов, построенных на различных уровнях абстракции, в том числе на уровне классического автомата и логической схемы. Мутационный подход позволяет единым образом строить тесты с гарантированной полнотой относительно любых мутаций в логической схеме. Еще одним аргументом для реализации новых методов построения тестов на основе логических схем для систем, поведение которых описано конечными системами переходов, является возможность их использования для неклассических автоматных моделей, таких как расширенные и временные автоматы, для которых методы синтеза проверяющих тестов с гарантированной полнотой развиты недостаточно хорошо, в то время как конечно-автоматный уровень достаточно хорошо исследован при построении таких тестов.

В диссертации исследованы свойства тестовых последовательностей, построенных для верификации логических компонентов, используемых в киберфизических и телекоммуникационных системах относительно наиболее распространенных моделей ошибок трех типов: одиночных константных неисправностей, одиночных неисправностей «перемычек» и одиночных неисправностей при замене одного вентиля схемы.

В работе сравниваются свойства тестов, построенных на основе кратчайших различающих последовательностей для обнаружения мутаций в описании поведения логических схем и таких последовательностей, сгенерированных псевдослучайным образом с использованием системы логического синтеза и верификации ABC. Эксперименты проводились для набора контрольных примеров (бенчмарков) b01-b10 из пакета ITC'99, которые представляют собой компоненты телекоммуникационных систем, спроектированных для различных приложений.

В ходе экспериментов сравнивались три подхода к построению так называемых полных тестов, т.е. тестов, обнаруживающих все возможные мутации заданного класса для эталонного описания логической схемы. В первом случае для каждой мутации находится обнаруживающая ее кратчайшая тестовая последовательность, во втором случае – тестовая последовательность генерируется псевдослучайным образом и, как правило, является более длинной, в третьем случае – первая часть последовательностей определенной длины генерируется случайным образом, и для каждой мутации, не обнаруженной данной последовательностью, достраиваются различающие последовательности. Наиболее эффективным является третий подход. Как показывают проведенные эксперименты, длина полного теста, построенного с использованием комбинированного алгоритма, оказывается минимальной.

Предполагая, что внесение неисправности в логическую схему отражается на ее функционировании, тесты, построенные на уровне конечного автомата, соответствующего логической схеме, должны обнаруживать такую неисправность. По результатам экспериментов полнота обнаружения различных видов неисправностей в логической схеме тестом, построенным на основе обхода графа переходов соответствующего ей конечного автомата, составляет более 90% для схем из пакета бенчмарков ITC'99 и логических схем, построенных по случайно сгенерированным автоматам с числом состояний от 5 до 20 и числом входов от 2 до 16. Таким образом, тесты, построенные на высоком уровне абстракции (по модели конечного автомата), имеют достаточно высокий процент покрытия относительно трех типов неисправностей в логических схемах, но, тем не менее, не достигают 100%.

В связи с этим в работе предлагается строить тесты для последовательных схем, которые бы являлись полными относительно неисправностей трех рассматриваемых типов логической схемы, а также относительно выходных ошибок в конечном автомате, соответствующем данной логической схеме. Если конечный автомат, построенный по логической схеме, оказывается слишком большим, то для построенного теста устанавливается, относительно какой автоматной модели неисправности тест является полным.

Пусть поведение логической схемы описывается автоматом  $\mathbf{S}$ , в котором множество состояний  $S$  есть подмножество всех двоичных наборов длины  $n$ , где  $n$  – число элементов задержки, и входной алфавит  $I$  содержит все двоичные наборы длины  $p$ , где  $p$  – число внешних входов логической схемы. Для каждой неисправности логической схемы построим соответствующий мутант; различающая последовательность для эталонной логической схемы и мутанта (если существует) строится с использованием одной из систем логического синтеза и верификации. Построенные последовательности собираются в единый тест  $TS$ , который можно дополнительно оптимизировать. Поведение логической схемы моделируется на последовательностях теста  $TS$  и строится соответствующий конечный автомат  $M_{LC}(TS)$ . В общем случае автомат  $M_{LC}(TS)$  детерминированный и частично определенный. Построим разбиение  $\pi$  множества  $S \times I$  на классы таким образом, что каждый класс множества содержит некоторую пару «состояние, входной символ», для которой определено поведение автомата  $M_{LC}(TS)$ . Рассмотрим модель неисправности  $\langle \mathbf{S}, \cong, FD_{\pi} \rangle$ , где  $\mathbf{S} = (S, I, O, next\_states, outs, so)$  – полностью определенный автомат-спецификация,  $\cong$  – отношение эквивалентности. Область неисправности  $FD_{\pi}$  содержит каждый полностью определенный автомат  $\mathbf{P}$  с входным алфавитом  $I$ , множество  $P$  состояний которого есть подмножество множества  $S$  и в котором возможны только выходные ошибки относительно эталонного автомата  $\mathbf{S}$ , причем для каждого класса  $b$  из  $\pi$  справедливо: либо выходной символ в автомате  $\mathbf{P}$  «правильный» для всех пар класса, либо для каждой пары класса в автомате  $\mathbf{P}$  есть ошибка в выходном символе.

**Теорема 1.** Проверяющий тест  $TS$  обнаруживает каждый автомат из класса  $FD_{\pi}$ , не эквивалентный эталонному автомату  $\mathbf{S}$ , т.е. является полным проверяющим тестом относительно модели неисправности  $\langle \mathbf{S}, \cong, FD_{\pi} \rangle$ .

**Следствие.** Если автомат  $M_{LC}(TS)$  является полностью определенным, то тест  $TS$  является полным относительно всех выходных неисправностей в автомате, соответствующем логической схеме.

Если автомат, соответствующий логической схеме, не удастся построить ввиду его громоздкости, то можно либо довольствоваться полнотой теста согласно теореме 1, либо достраивать тест до покрытия тех

переходов, которые достаточно просто промоделировать. При этом, согласно проведенным экспериментам, после увеличения длины теста в 2-4 раза можно полагать, что большое количество переходов в автомате окажется покрытым.

В работе приводятся экспериментальные результаты по оценке времени генерации теста для схем конечно-автоматным методом и мутационным способом, показывающие намного бóльшую масштабируемость конечно-автоматного способа при использовании обхода графа переходов и сохраняющие полноту построенных тестов порядка 90% относительно неисправностей в соответствующей логической схеме. Для построения полных тестов можно использовать тест, построенный по W-методу для увеличения числа состояний на 20%. Однако для автоматов с числом состояний более 1000 и числом входных символов более 4 при использовании W-метода даже в предположении увеличения числа состояний на единицу тесты строятся несколько часов. Поэтому если логическая схема имеет более 10 элементов задержки, то можно использовать подход, интегрирующий методы синтеза тестов по модели конечного автомата и методы синтеза теста по логической схеме для построения качественного теста: тест, построенный на первом шаге обходом графа переходов автомата, соответствующего логической схеме, на следующем шаге дополняется последовательностями, обнаруживающими мутации логической схемы, не обнаруженные на первом шаге.

Таким образом, результаты данной главы подтверждают «равноправность» использования обеих уровней абстракции при построении проверяющих тестов: конечных автоматов и логических схем; кроме того, предлагается способ интеграции тестов, построенных на различных уровнях абстракции, с учетом достоинств и недостатков синтеза тестов на каждом из уровней.

В **третьей** главе рассматривается алгоритм синтеза тестов для еще одного уровня абстракции, а именно, для уровня временного автомата, который является расширением модели классического автомата. Результаты, представленные в данной главе, опубликованы в работах [2, 3, 6, 9, 10].

Поскольку большинство систем строятся как композиции более простых в некотором смысле подсистем, на первом этапе исследуется проблема построения параллельной композиции для автоматов с таймаутами. Операция параллельной композиции достаточно часто используется при описании поведения взаимодействующих компонентов программных и киберфизических систем. В этом случае автоматы-компоненты работают в режиме диалога при наличии медленной внешней среды, т.е. когда входное воздействие на систему подается только после получения реакции системы на предыдущее входное воздействие. При решении задач анализа, в том

числе для построения проверяющих тестов для системы взаимодействующих детерминированных конечных автоматов с использованием классических методов, желательно, чтобы поведение такой композиции описывалось также детерминированным автоматом.

Известно, что, если компоненты композиции являются детерминированными полностью определенными автоматами, то поведение параллельной композиции описывается полностью определенным детерминированным автоматом при наличии медленной внешней среды и отсутствии осцилляций, т.е. бесконечного диалога между компонентами. В этом случае состояния автомата-композиции обычно представляют собой пару состояний компонент. Задача построения композиций автоматов с таймаутами рассматривалась в ряде работ, где, в частности, предлагается перейти от временных автоматов к соответствующим полуавтоматам, представляющим регулярные языки, и построить композицию путем пересечения полуавтоматов. Однако в диссертации показано, что данный подход становится неадекватным в том случае, когда временная переменная может принимать не целочисленные значения.

**Теорема 2.** Множество детерминированных конечных автоматов с таймаутами не является замкнутым относительно операции параллельной композиции.

Причиной утверждения теоремы 2 является тот факт, что компонента, не взаимодействующая с внешней средой, «накапливает» вещественное значение временной переменной и в зависимости от момента поступления внешнего входного воздействия может «перейти» или «не перейти» целочисленный порог, изменяющий поведение компоненты. Последнее согласуется с результатами в области временных полуавтоматов, где показано, что не для всякого временного полуавтомата с несколькими временными переменными существует эквивалентный полуавтомат с одной временной переменной.

Тем не менее, описание композиции автоматов с таймаутами в виде детерминированного автомата возможно в случае, когда временная переменная принимает только целочисленные значения, а также для определенных классов автоматов.

**Теорема 3.** Параллельная композиция автоматов с таймаутами  $S$  и  $P$  может быть описана детерминированным временным автоматом, если в компоненте  $S$  не существует перехода  $(s, i, s', o) \in h_S$  (в компоненте  $P$  не существует перехода  $(p, i, p', o) \in h_P$ ), где  $i$  и  $o$  – внешние входной и выходной символы, и в построенной параллельной композиции в каждом состоянии входной таймаут больше, чем выходные задержки для обработки каждого входного символа.



Если поведение параллельной композиции автоматов с таймаутами описывается детерминированным автоматом, то тесты для такой композиции можно строить на основе логической схемы, построенной по конечно-автоматной абстракции. В этом случае используется мутационный подход к синтезу тестов для логических схем, описанный в главе 2 диссертации, где множество мутаций логической схемы достаточно широкое.

Вышеописанный подход был использован при синтезе тестов для микроконтроллерной системы<sup>1</sup>, поведение которой описывается автоматом с таймаутами. Система представляет собой реализацию коммутирующего генератора на микроконтроллере ATmega8 и используется для управления CCD-камерой при исследовании колебаний биморфного пьезоэлемента в виде пластины. Для спецификации системы управления CCD-камерой была выбрана модель конечного автомата с таймаутами. Таймауты помогают описать состояния системы, в которых происходит превышение периода входных импульсов, соответствующего определенной частоте. По временному автомату с таймаутами была построена конечно-автоматная абстракция. По конечно-автоматной абстракции временного автомата с таймаутами была построена логическая схема. С помощью мутационного подхода был построен полный тест относительно трех типов неисправностей в логических схемах. Построенный тест содержал 3201 входной символ и показал высокую полноту покрытия наиболее распространенных ошибок в исходном коде программы для микроконтроллера при проведении мутационного тестирования.

В **четвертой** главе исследуется задача синтеза логических схем для проверки нефункциональных требований, а именно определения характеристик доверия к приложениям, выполняющимся на устройствах с ограниченными вычислительными возможностями в интернете вещей. Результаты, представленные в главе, опубликованы в работах [8, 11].

В настоящее время, в связи с интенсивным развитием в сфере информационных технологий такого направления как интернет вещей (Internet of Things, IoT), растет использование устройств с ограниченными вычислительными возможностями. Реализация на данных устройствах методов классификации, основанных на технологиях машинного обучения, вызывает большой интерес, так как данные устройства активно взаимодействуют друг с другом и часто возлагают друг на друга выполнение критических операций. В настоящей главе исследуются вопросы использования логических схем для интеграции в устройства с ограниченными вычислительными возможностями в интернете вещей

---

<sup>1</sup> Работы проводились совместно с АО «НВП «Топаз»», о чем в диссертации представлен акт внедрения.

специального модуля для проверки нефункциональных требований к другим устройствам в сети или к приложениям, выполняющимся на данном устройстве, а именно для оценки характеристик доверия к приложениям.

В литературе описаны различные методики машинного обучения, применяемые для решения различных задач классификации, например, искусственные нейронные сети (Artificial Neural Network, ANN), системы опорных векторов (Support Vector Machine, SVM). Различными исследователями предложены алгоритмы для обучения систем опорных векторов, нейронных сетей и других моделей с высокой точностью предсказания. Главным недостатком подобных моделей в случае их применения в устройствах с ограниченными вычислительными возможностями является их высокая требовательность к вычислительным ресурсам устройства. Поэтому в диссертации предложено использовать логические схемы, которые строятся на основе обученной с достаточной точностью модели машинного обучения. Поскольку число параметров, которые существенно влияют на оценку характеристик доверия, в устройствах с ограниченными вычислительными возможностями, как правило, не слишком большое, использование логических схем оказалось эффективным.

При реализации модели машинного обучения необходимо построить логическую схему по множеству значений параметров и сопоставленных им значений меток. Первый шаг заключается в кодировании каждого набора значений параметров булевым вектором. Каждому такому булеву вектору ставится в соответствие булев вектор, кодирующий значение метки для соответствующего набора значений параметров. После обучения модели на всех значениях параметров или в заданной области полученные булевы векторы формируют таблицу истинности для системы булевых функций, по которой и нужно синтезировать логическую схему любым из известных методов. В данной работе для синтеза логической схемы используется система логического синтеза и верификации ABC. Комбинационная логическая схема может быть реализована аппаратно и далее интегрирована в устройство с ограниченными вычислительными возможностями. Необходимо отметить, что предлагаемый алгоритм может использоваться, вообще говоря, для работы с любой предсказывающей моделью с разумно ограниченным множеством параметров. В работе также рассматривается композиция логических схем для случая, когда имеются несколько логических схем, построенных по моделям машинного обучения, обученных для классификации данных из пересекающихся множеств. Такая композиция помогает повысить точность классификации значений по сравнению с исходными моделями.

В работе представлены результаты серии экспериментов, которые показывают, что такие модели машинного обучения как искусственные нейронные сети и системы опорных векторов при вычислении определенных характеристик доверия проигрывают в производительности логической схеме, которая моделируется программно. Для моделирования логической схемы была использована программа ABC 1.01 (команда *sim*). Моделирование схемы на входных наборах производилось 10000 раз для измерения среднего времени работы, которое составило 20 мс. Таким образом, в среднем на обработку одного входного набора уходит 0,002 мс. При использовании среды GNU Octave 3.4.3 с пакетом LibSVM 3.22 время, затрачиваемое на симуляцию SVM на том же самом входном наборе, составило 0,95 мс. Таким образом, производительность логической схемы превосходит модель SVM в 424 раза.

Во второй серии экспериментов использовались программные реализации нейронной сети на языке C и программные реализации логической схемы на языке C, получаемые по Verilog описанию схемы. В ходе эксперимента было сгенерировано несколько нейронных сетей с различным количеством нейронов в скрытом слое. Каждая нейронная сеть моделировалась на всех входных наборах и строилась соответствующая логическая схема, которая далее представлялась программой на языке C. В таблице 1 представлено среднее время, затрачиваемое нейронной сетью и соответствующей логической схемой на обработку одного входного набора.

Как показывают проведенные эксперименты, во многих случаях даже оптимизированные реализации моделей машинного обучения значительно проигрывают оптимизированным программным реализациям логической схемы. Более того, учитывая подход, по которому синтезируется схема, время предсказания логической схемы не зависит от сложности модели, по которой синтезируется схема.

Таблица 1 – Время обработки одного входного набора нейронной сетью и логической схемой

Скрытые нейроны	Время работы ANN, мс	Время работы схемы, мс	Ускорение схемы
10	1,89	1,55	1,22
20	2,02	1,56	1,29
30	2,14	1,55	1,38
40	2,23	1,6	1,39
50	2,25	1,57	1,43

Окончание таблицы 1

Скрытые нейроны	Время работы ANN, мс	Время работы схемы, мс	Ускорение схемы
60	2,42	1,55	1,56
70	2,46	1,56	1,58
80	2,62	1,56	1,68
90	2,72	1,54	1,76
100	2,80	1,45	1,93

Значения переменных в исходном коде программы могут отражать показатели доверия к данному приложению, однако зависимость является достаточно сложной, и поэтому характеристики доверия достаточно часто оцениваются с использованием моделей машинного обучения. Соответственно, рассмотренный в предыдущем разделе подход можно использовать для оценивания характеристик доверия к приложению, выполняемому на некотором устройстве, в том числе в сети интернета вещей. Одним из наиболее важных аспектов, влияющих на характеристики доверия, является объем ресурсов, потребляемых данным приложением, и, вообще говоря, можно использовать различные параметры для оценки потребления ресурсов. В работе рассматриваются 5 таких параметров: размер кучи (память, выделяемая приложением динамически), использование стека (память, выделяемая приложением статически), использование ЦПУ (центральное процессорное устройство), использование диска, потребление энергии.

Были проведены эксперименты с системой удаленного мониторинга температуры по протоколу Ethernet, реализованной на микроконтроллере LPC4088. Запущенное на микроконтроллере приложение постоянно считывает значения датчика температуры и постоянно прослушивает HTTP порт на предмет входящих соединений, выполняя функционал HTTP-сервера. При появлении HTTP запроса, приложение отправляет HTTP-клиенту сообщение с текущим значением температуры.

Для рассматриваемого приложения были определены референсные значения потребляемых параметров. На основе референсных значений диапазон допустимых значений рассматриваемых параметров был разбит на интервалы, соответствующие обычному режиму работы приложения и режимам с отклонениями от обычного. Для значений указанных выше параметров предполагалось определить уровень доверия, определяемый двумя значениями 0 (не доверять) и 1 (доверять). Множество всевозможных входных наборов значений параметров составило 108 наборов, на каждом из которых значение характеристики доверия определено. На половине наборов

из исходного множества была обучена нейронная сеть, и по нейронной сети была построена логическая схема. Значения, выдаваемые схемой на второй половине наборов, сравнивались со значениями в исходном множестве данных. Результаты эксперимента показали высокую точность модели логической схемы, построенной по нейронной сети, а также превосходство по скорости обработки входных данных.

Таким образом, глава 4 содержит предложенную методику реализации моделей машинного обучения в виде логической схемы, воспроизводящей поведение сложной модели машинного обучения. Такая схема может использоваться как масштабируемое устройство для вычислительно сложного предсказания/классификации при интеграции в устройство с ограниченными вычислительными возможностями, работающего в интернете вещей. Предложенная методика является общей и подходит для реализации любой модели машинного обучения на устройстве с ограниченными вычислительными возможностями.

В **заключении** приводятся основные результаты, полученные в диссертации и направления для дальнейших исследований.

На основе логических схем предложены алгоритмы синтеза и оптимизации тестов для проверки функциональных свойств компонентов, поведение которых описывается моделью классического автомата или автомата с таймаутами. Алгоритмы опробованы на ряде примеров, в том числе на синтезе тестов для микроконтроллеров, используемых в киберфизических и телекоммуникационных системах. Проведенные эксперименты показали достаточно высокое качество построенных тестов относительно распространенных ошибок в исходных кодах программ для микроконтроллеров, в том числе для случаев, когда поведение тестируемого компонента существенно зависит от временных аспектов.

Предложена методика реализации моделей машинного обучения в виде логической схемы для последующей интеграции в устройство с ограниченными вычислительными возможностями для оценки характеристик доверия к приложениям, выполняющимся на устройстве, что, в частности, необходимо при функционировании таких устройств в интернете вещей. Экспериментальные результаты показали превосходство по скорости работы (в некоторых случаях в сотни раз) реализации в виде логических схем таких моделей машинного обучения как система опорных векторов и нейронная сеть типа перцептрон по сравнению с исходными программными реализациями данных моделей, в том числе на примере задачи определения характеристик доверия к приложению для удаленного мониторинга температуры через интернет, выполняющемуся на микроконтроллере с ограниченными вычислительными возможностями. Кроме того, реализованные на ПЛИС соответствующие логические схемы позволяют

получить большое преимущество в скорости обработки данных. Следует отметить, что предложенная методика может быть использована для проверки / контроля других нефункциональных требований, таких как безопасность, качество и др.

Для дальнейших исследований представляет интерес использование результатов главы 3 для построения тестов с гарантированной полнотой для расширенных автоматов, которые являются наиболее близкой моделью по уровню абстракции к программным системам; разработка автоматических методов синтеза тестов с гарантированной полнотой для расширенных автоматов является актуальной задачей. Другой интересной задачей является реализация в виде логической схемы моделей машинного обучения без учителя или моделей, обучающихся с подкреплением, например, искусственных нейронных сетей, которые в настоящее время набирают популярность в различных областях обработки данных.

### ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

*Статьи в журналах, включенных 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:*

1. **Лапутенко А. В.** Синтез тестов для цифровых систем на высоком и низком уровнях абстракции / А. В. Лапутенко, Е. М. Винарский // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2019. – № 47. – С. 110–117. – DOI: 10.17223/19988605/47/13. – 0,54 / 0,27 а.л.

*Web of Science:* **Laputenko A. V.** Deriving tests for digital circuits at lower and higher abstraction levels / A. V. Laputenko, E. M. Vinarsky // Vestnik Tomskogo gosudarstvennogo universiteta-Upravlenie vychislitel'naja tehnika i informatika – Tomsk state university journal of control and computer science. – 2019. – № 47. – P. 110–117.

2. Твардовский А. С. О возможностях автоматного описания параллельной композиции временных автоматов / А. С. Твардовский, **А. В. Лапутенко** // Труды Института системного программирования РАН. – 2018. – Т. 30, № 1. – С. 25–40. – DOI: 10.15514/ISPRAS-2018-30(1)-2. – 0,78 / 0,39 а.л.

3. Громов М. Л. Использование временных автоматов при тестировании киберфизических систем / М. Л. Громов, Н. В. Евтушенко, А. В. Лапутенко // Известия высших учебных заведений. Физика. – 2016. – № 12. – С. 174–176. – 0,18 / 0,06 а.л.

*в переводной версии, входящей в Web of Science:*

Gromov M. L. Testing cyber-physical systems using timed finite state machines / M. L. Gromov, N. V. Yevtushenko, **A. V. Laputenko** // Russian

Physics Journal. – 2017. – Vol. 59, is. 12. – P. 2181–2182. – DOI: 10.1007/s11182-017-1033-7.

4. **Лапутенко А. В.** Обработка экспериментальных данных при верификации компонентов физических систем: оценка качества тестовых последовательностей / А. В. Лапутенко, Х. Е. Лопез, Н. В. Евтушенко // Известия высших учебных заведений. Физика. – 2017. – № 11. – С. 174–176. – 0,38 / 0,13 а.л.

*в переводной версии, входящей в Web of Science:*

**Laputenko A. V.** Verifying Digital Components of Physical Systems: Experimental Evaluation of Test Quality / A. V. Laputenko, J. E. López, N. V. Yevtushenko // Russian Physics Journal. – 2017. – Vol. 60, is. 11. – P. 2012–2018. – DOI: 10.1007/s11182-018-1317-6.

*Статьи в сборниках материалов конференций, представленных в изданиях, входящих Web of Science и Scopus:*

5. López J. On the Fault Coverage of High-level Test Derivation Methods for Digital Circuits / J. López, E. Vinarsky, **A. Laputenko** // International Conference and Seminar of Young Specialists on Micro-Nanotechnologies and Electron Devices (EDM 2017) : proceedings of 18th International Conference. Erlagol, Altai Republic, Russian Federation, June 29 – July 03, 2017. – Novosibirsk, 2017. – P. 184–189. – 0,67 / 0,22 а.л. (*Web of Science*).

6. **Laputenko A.** Testing Microcontroller Based Physical Systems Using Finite Transition Models / A. Laputenko, T. Petukhov, N. Vasnev // Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM 2018) : proceedings of 19th International Conference. Erlagol, Altai Republic, Russian Federation, June 29 – July 03, 2018. – Novosibirsk, 2018. – P. 203–206. – DOI: 10.1109/EDM.2018.8435029.– 0,33 / 0,11 а.л. (*Scopus*).

7. Vinarskii E. Testing Digital Circuits: Studying the Increment of the Number of States and Estimating the Fault Coverage / E. Vinarskii, **A. Laputenko**, J. López, N. Kushik // Conference of Young Specialists on Micro-Nanotechnologies and Electron Devices (EDM 2018) : proceedings of 19th International Conference. Erlagol, Altai Republic, Russian Federation, June 29 – July 03, 2018. – Novosibirsk, 2018. – P. 220–224. – DOI: 10.1109/EDM.2018.8435051.– 0,44 / 0,11 а.л. (*Scopus*).

8. López J. Scalable Supervised Machine Learning Apparatus for Computationally Constrained Devices / J. López, **A. Laputenko**, N. Kushik, N. Yevtushenko, S. Torgaev // ICSOFT 2018 : proceedings of the 13th International Conference on Software Technologies. Porto, Portugal, July 26–28, 2018. – 2019. – P. 518–528. – DOI: 10.5220/0006908905180528. – 1,15 / 0,23 а.л. (*Scopus*).

9. **Laputenko A.** Logic Circuit Based Test Derivation for Microcontrollers / A. Laputenko // Conference of Young Specialists on Micro-Nanotechnologies and Electron Devices (EDM 2019) : proceedings of 20th International Conference. Erlagol, Altai Republic, Russian Federation, June 29 –

July 03, 2019. – Novosibirsk, 2019. – P. 70–73. – DOI: 10.1109/EDM.2019.8823364.– 0,42 а.л. (*Scopus*).

*Публикации, в прочих научных изданиях:*

10. **Лапутенко А. В.** Реализация и тестирование системы сигнализации на базе микроконтроллера STM32F407VG / А. В. Лапутенко, М. Л. Громов, С. Н. Торгаев // Известия высших учебных заведений. Физика. – 2016. – Т. 59, № 8/2. – С. 61–64. – 0,23 / 0,08 а.л.

11. **Лапутенко А. В.** Сравнение производительности различных реализаций самообучающихся моделей на примере задачи классификации / А. В. Лапутенко, С. Н. Торгаев // Новые информационные технологии в исследовании сложных структур : материалы 12-й Российской конференции с международным участием. пос. Катунь Алтайский край, 04–08 июня 2018 г. – Томск, 2018. – С. 71–72. – 0,07 / 0,03 а.л.



Издание подготовлено в авторской редакции.  
Отпечатано на участке цифровой печати  
Издательства Томского государственного университета  
Заказ № 7299 от «21» апреля 2021 г. Тираж 110 экз.  
г. Томск Московский тр.8, тел. 53-15-28  
[publish.tsu.ru](http://publish.tsu.ru)