Санкт-Петербургский политехнический университет

Петра Великого

Институт кибербезопасности и защиты информации

На правах рукописи

Крундышев Василий Михайлович

АВТОМАТИЗИРОВАННАЯ СИСТЕМА АНАЛИЗА КИБЕРУГРОЗ В КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЕ

Направление подготовки: 10.06.01 Информационная безопасность

Направленность: 10.06.01_01 «Методы и системы защиты информации, информационная безопасность»

НАУЧНО-КВАЛИФИКАЦИОННАЯ РАБОТА

Автор работы: Крундышев В.М.
Научный руководитель: профессор,
д.т.н., Калинин М.О.

Санкт-Петербург – 2021

Оглавление

ВВЕДЕНИЕ

Актуальность темы исследования. Традиционные стратегии обеспечения кибербезопасности не могут быть напрямую применены при построении и оценки цифровых критических инфраструктур. Это объясняется их новизной, неоднородностью, высокой степенью масштабируемости, динамикой смены топологии и состояния безопасности, и наличием огромного количества горизонтальных связей. Для управления кибербезопасностью таких сетей впервые предлагается использовать разработанный метод на основе технологии программно-конфигурируемых сетей. Разработанный метод выявления кибератак, направленных на крупные динамические цифровые инфраструктуры с использованием методов искусственного интеллекта, позволяет обеспечивать динамическую защиту в условиях текущего ландшафта киберугроз. Помимо этого, адекватный и точный анализ киберрисков безопасности для структурных компонент динамических сетей является крайне важной задачей, точные и объективные результаты которой позволяют строить адекватные системы защиты и оценивать их эффективность. Для этого требуется создавать новые методы, либо совершенствовать существующие. Для решения этой задачи впервые предлагается использовать математический аппарат, который заключается в создании новой методики количественной оценки киберрисков, учитывающего типизацию и самоорганизацию цифровых критических инфраструктур, а также их взаимное влияние друг на друга.

Степень разработанности темы исследования. Проблеме обеспечения информационной безопасности объектов критической информационной инфраструктуры посвящено множество исследований российских и иностранных учёных, таких как Котенко И. В., Петренко С. А., Аветисяна А. И., Гросена Дж., Грейвса А., Вейна Г. и других. Предлагаемый автором подход к обнаружению киберугроз, направленных на объекты критической информационной инфраструктуры, основан на автоматическом выбора

интеллектуальных методов обеспечения безопасности. Данный подход позволяет динамически в зависимости от текущих сетевых характеристик осуществлять выбор наиболее эффективного средства обнаружения вредоносного воздействия.

Цель исследования. Целью исследования является обеспечение информационной безопасности критической инфраструктуры на основе рационального многокритериального выбора защитных механизмов. Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Определение особенностей объектов критической информационной инфраструктуры и применяемых в них механизмов обеспечения безопасности.
2. Разработка модель типового объекта критической информационной инфраструктуры.
3. Разработка подхода к выбору защитных мер, основанного на нейронечеткой модели.
4. Разработка метода выбора вычислительных ресурсов платформы эластичных вычислений.
5. Разработка метода оценки защищенности активов критической информационной инфраструктуры и выработки контрмер по снижению киберрисков.

Научная новизна

Предложен подход, который в зависимости от текущего состояния контролируемой сети на основе динамической многокритериальной оценке, определял наиболее эффективный механизм защиты.

Теоретическая и практическая значимость

Теоретическую значимость работы составляют предложенные методы обнаружения кибервоздействий на объекты КИИ. Разработана модель типового объекта КИИ с использованием системы «хищник-жертва». Предложен подход к автоматическому выбору защитных механизмов.

Практическая значимость работы определяется возможностью использования предложенных методов для обеспечения защищенность объектов критической информационной инфраструктуры.

Апробация работы

Промежуточные результаты исследований и научных разработок обсуждались на научной конференциях: 1st IEEE International Conference on Industrial Cyber-Physical Systems, ICPS 2018; ITMO UniversitySaint Petersburg; Russian Federation; 15-18 May 2018, 11th International Conference on Security of Information and Networks, SIN 2018; Cardiff; United Kingdom; 10-12 September 2018, XIII Всероссийское совещание по проблемам управления. Россия, Москва, ИПУ РАН, 17-20 июня 2019, IEEE International Conference on Industrial Cyber-Physical Systems, г. Тайпэй, Тайвань, 6-9 мая 2019 г., 7th International Black Sea Conference on Communications and Networking, г. Сочи, 3-6 июня 2019 г., 2020 International Russian Automation Conference, RusAutoCon 2020, Sochi, 6 September 2020 - 12 September 2020., 29-я конференция «Методы и технические средства обеспечения безопасности информации», Санкт-Петербург, 28-30 сентября 2020., 13th International Conference on Security of Information and Networks, SIN 2020, Virtual, Online, 4 November 2020 - 6 November 2020.

Структура и объем диссертации.

Диссертация состоит из введения, четырёх глав, заключения, списка использованных источников из 30 наименований.

Общий объем работы составляет 73 страницы, в том числе 31 рисунок, 7 таблиц.

Во введении обоснована актуальность выбранной темы диссертационного исследования и поставлена цель и определены основные задачи, необходимые для достижения поставленной цели, раскрыта научная новизна исследования, его практическая значимость.

В первой главе исследована специфика объектов критической информационной инфраструктуры, а также определены основные киберугрозы.

Во второй главе представлена разработанная модель типового объекта критической информационной инфраструктуры с использованием модифицированной модели «хищник-жертва»

В третьей главе изложен подход автоматического выбор защитных механизмов для обнаружения киберугроз на основе аппарата нейронечеткой логики.

В четвёртой главе представлены разработанные интеллекутальные методы по обнаружению кибервоздействий, включая нейросетевой анализ, квантовое машинное обучение и выравнивание биологических последовательностей.

В пятой главе представлен разработанный метод анализа рисков информационной безопасности, основанный на количественной оценке.

В заключении приведены основные результаты и выводы, полученные автором в ходе выполнения работы.

1 ИССЛЕДОВАНИЕ ОБЪЕКТОВ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

Цифровое производство подразумевает применение интегрированных компьютерных систем для производственных услуг, цепочек поставок и прочих процессов [1]. Переход к цифровому производству стал более массовым в связи с ростом количества и качества компьютерных систем на производственных предприятиях. Поскольку на производственных предприятиях стало использоваться все больше автоматизированных инструментов, появилась необходимость смоделировать и проанализировать все станки, инструменты и исходные материалы с целью оптимизировать процесс производства. Целью цифровых производственных технологий является объединение систем и процессов во всех областях производства для создания интегрированного подхода к производству от дизайна до обслуживания конечного продукта [2].

Моделируя производственные процессы, становится возможным повысить качество принятия производственных решений, уменьшить расходы, сократить время выхода продукта на рынок и создать производственный процесс, который объединяет цифровые инструменты с физическим исполнением производства.

Такая концепция позволяет собирать и анализировать данные в течение всего жизненного цикла продукта и отправлять данные о потребителях менеджерам по продукции, чтобы предвидеть спрос и любые текущие требования к обслуживанию продуктов.

1.1 Индустрия 4.0

Индустрия 4.0 – это текущая трансформация традиционных промышленных и производственных практик с использованием новейших интеллектуальных технологий. В первую очередь это касается использования крупномасштабных коммуникаций между машинами (machine to machine communication, M2M), развертывания Интернета вещей (IoT) для обеспечения

повышенной автоматизации, а также "умных" машин, способных анализировать и диагностировать проблемы без вмешательства человека [3].

Одним из примеров цифровой трансформации являются облачные вычисления. Облачные вычисления позволяют уменьшить зависимость от пользовательского оборудования, полагаясь на облачные сервисы. Некоторые из этих цифровых решений расширяют возможности традиционных программных продуктов (например, Office 365 от Microsoft), в то время как другие полностью основаны на облаке (например, Google Docs).

Цифровая трансформация идет по всему миру, но не везде одинаковыми темпами. Согласно индексу оцифровки промышленности (McKinsey Global Institute's Industry Digitization Index [5]), который создан для отслеживания прогресса в области цифровизации бизнеса, на 2016 год Европа использовала 12% своего цифрового потенциала, в то время как США – 18%.

Процесс цифровизации можно разделить на четыре части [6]:

1. Этап производства. Данный компонент измеряет степень, в которой предприятия оцифровывают свои внутренние процессы. Это включает в себя долю предприятий, использующих облачные вычисления, big data, автоматизированные системы, операционные системы с открытым исходным кодом.

2. Восходящая фаза цепочки поставок. Этот компонент изучает, насколько предприятие оцифровывает свою практику связи с внешними поставщиками. Сюда относятся такие меры, как доля предприятий, совершающих покупки онлайн у поставщиков, или степень использования предприятием Интернета для доступа к внешней информации.

3. Нисходящая фаза цепочки поставок. Это мера того, сколько предприятий оцифровывают свою практику связи с клиентами – будь то обычные потребители, другие предприятия или, например, правительства. Так же сюда относится доля предприятий, продающих онлайн, или доля предприятий, предоставляющих возможность онлайн-заказа или бронирования для клиентов.

4. Цифровая инфраструктура. Этот последний компонент рассматривает то, насколько предприятиям удалось создать инфраструктуру, которая поддерживает цифровизацию этапов производства. В частности, рассматриваются меры подключения, такие как широкополосный доступ, и доля сотрудников, которым предоставляется портативное устройство, позволяющее получать доступ в Интернет.

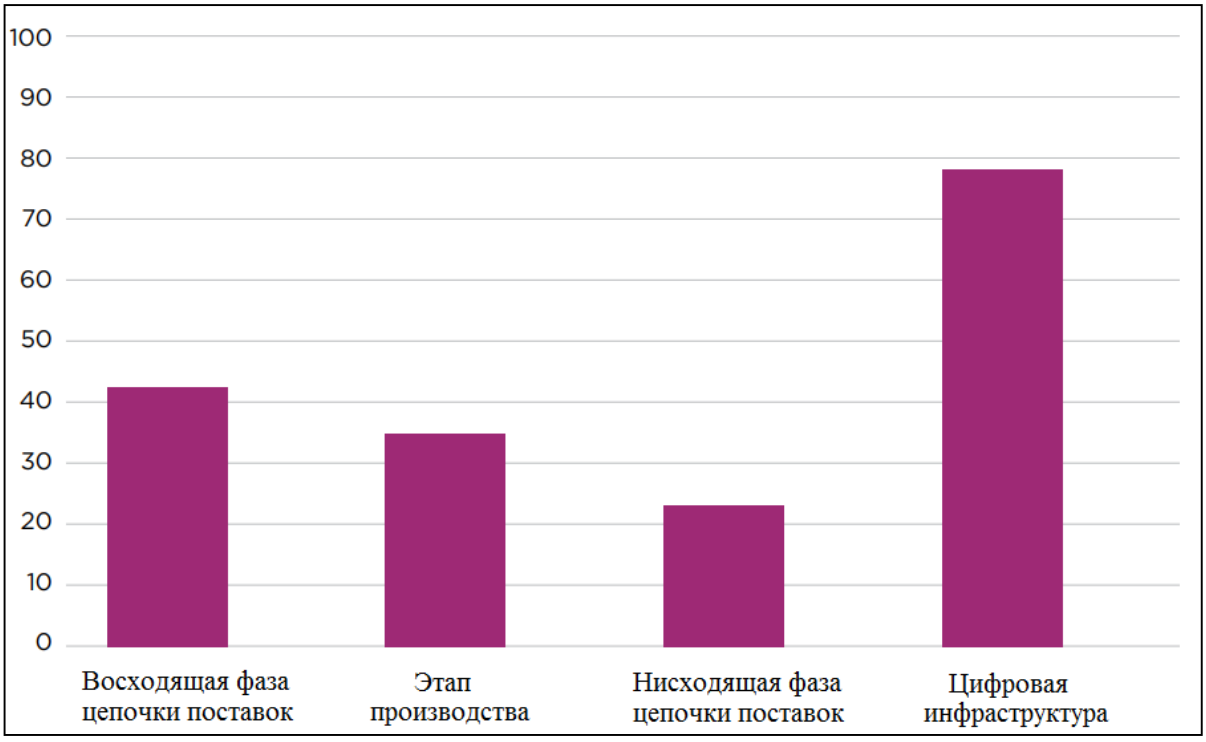

Рисунок 1 – Оценка по индексу цифровизации промышленности (IDI), средняя по всем отраслям на 2018 год [6]

Как видно из рисунка 1, процесс цифровизации по четырем компонентам идет неравномерно. Уровень цифровой инфраструктуры составляет 79 (где 100 соответствует полной оцифровке), в то же время нисходящая фаза цепочки поставок реализована только на 23 процента. Тем не менее даже этот показатель растет с каждым годом (в 2016 году этот показатель был равен 22).

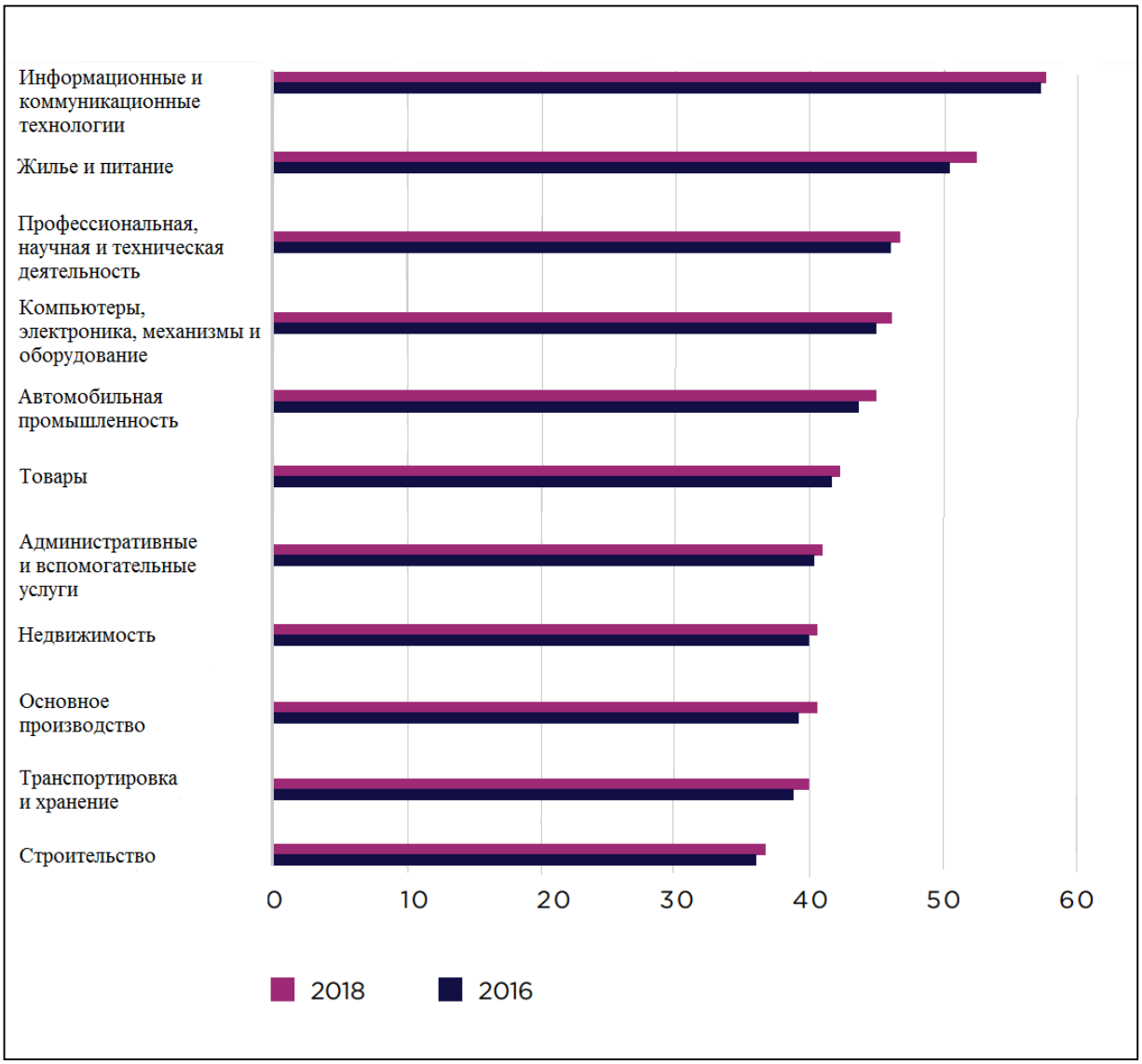

Рисунок 2 – Оценка по индексу цифровизации промышленности (IDI) по отраслевой группе [6]

Рисунок 2 наглядно демонстрирует увеличение индекса цифровизации по всем отраслям, ожидается, что данный индекс будет постепенно увеличиваться в сторону значения 100 в течение следующего десятилетия. Это означает, что все сферы деятельности человека с каждым днем становятся все более цифровыми.

1.2 Критическая информация инфраструктура

Критическая информационная инфраструктура (КИИ) — это информационные системы, информационно-телекоммуникационные сети, автоматизированные системы управления, а также сети электросвязи, используемые для организации их взаимодействия.

Цифровые решения в промышленности способны повысить эффективность, увеличить производительность и сделать производство более гибким.

Цифровое предприятие представляет собой предприятие, использующее информационные технологии на всех этапах своей деятельности. Это то предприятие, которое переходит на новый, цифровой уровень промышленности благодаря процессу цифровой трансформации.

На рисунке 3 представлена схема типового цифрового предприятия:

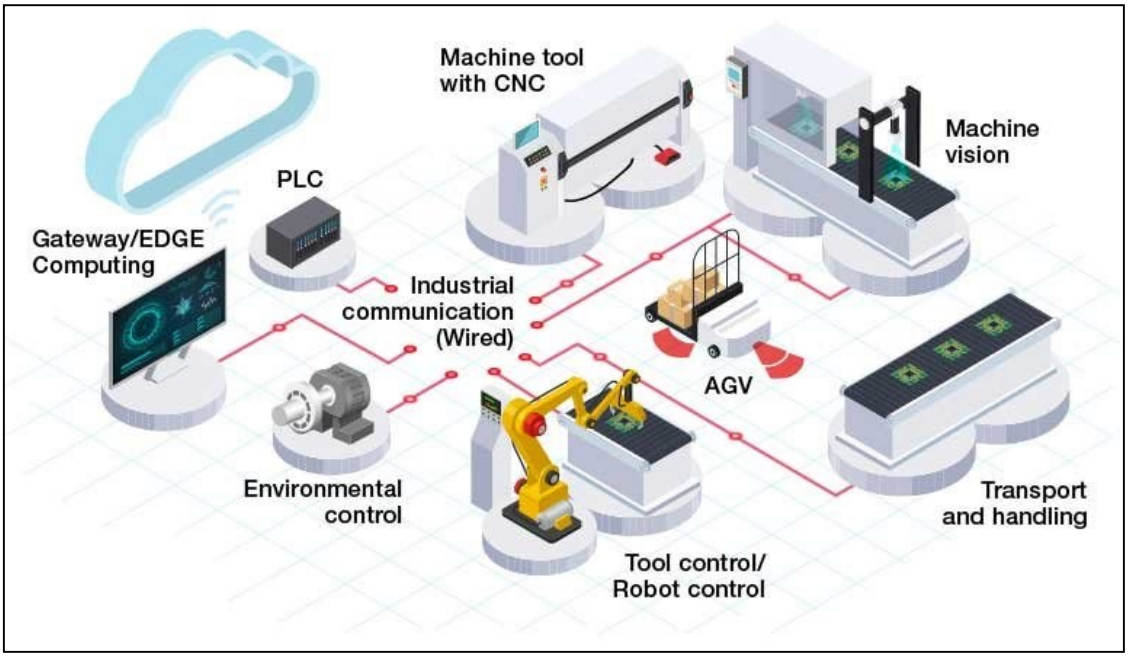

Рисунок 3 – Цифровое предприятие [7]

"Умная" фабрика предполагает использование интеллектуальных машин, датчиков и инструментов для обеспечения обратной связи о процессах и технологиях производства в реальном времени. Объединяя операционные и информационные технологии, становится возможным добиться лучшей

видимости процессов производства, контроля и оптимизации для повышения производительности.

В основе индустрии 4.0 лежит четыре основных принципа, которые помогают предприятиям в определении и реализации методов цифровой трансформации [4]:

1. Взаимосвязь: способность машин, устройств, датчиков и людей связываться между собой и общаться друг с другом через Интернет вещей (IoT) или через Интернет людей (IoP).

2. Информационная прозрачность: прозрачность, обеспечиваемая технологиями индустрии 4.0, предоставляет операторам огромное количество полезной информации, необходимой для принятия подходящих решений. Взаимосвязанность компонент позволяет собирать большие объемы данных со всех точек производственного процесса, тем самым улучшая функциональность и определяя ключевые области, которые могут принести пользу при их развитии и улучшении.

3. Техническая помощь: способность систем помогать людям в принятии решений и решении проблем, а также способность помогать в выполнении задач, которые являются слишком сложными или небезопасными для человека.

4. Децентрализованные решения: способность киберфизических систем принимать решения самостоятельно и выполнять свои задачи настолько автономно, насколько это возможно. Только в случае исключений, помех или противоречивых целей, задачи делегируются на более высокий уровень.

Таким образом, отличительной особенностью цифрового предприятия является то, что внутри него создается единое информационное пространство, в котором компоненты оперативно обмениваются информацией, а системы функционируют как можно более автономно.

Однако данная особенность, являясь совершенно новой концепцией производства, неизбежно влечет за собой появление различных угроз информационной безопасности: огромная информационная система содержит

множество различных компонент, каждая из которых обладает своими информационными ресурсами, обращается к ресурсам других компонент, если эти компоненты взаимосвязаны, обменивается данными с операторами и т.д. В связи с этим появляются новые требования к защите предприятия от кибератак.

1.3 Угрозы безопасности

Как было сказано выше, цифровое производство направлено на создание продуктов с более высоким качеством и более низкими затратами на их производство путем интеграции промышленного Интернета вещей, аналитики big data и т.д. Поскольку производственные машины все больше оснащаются датчиками и сенсорами, а также подключаются через беспроводные сети или проводной Ethernet, цифровые системы становятся намного более доступными, чем когда-либо.

Компании, которые придерживаются концепции цифрового производства крайне подвержены угрозам информационной безопасности, потому что они используют самый важный ресурс цифровой эпохи – данные. Для предприятия цифрового производства проблема кибербезопасности, связанная с данными может быть выражена следующим образом:

- конфиденциальность. Включает в себя кражу технических и деловых данных, а также интеллектуальную собственность;
- целостность. Включает в себя изменение данных, которое влечет за собой изменения в производственном процессе и изменение продукта;
- доступность. Повреждение или отказ в управлении процессом, которое может привести к сбою или остановке производственных операций.

Согласно анализу, проведенному в 2012 году [8], производственный/промышленный сектор оказался наиболее пострадавшим сектором с точки зрения кибератак (рисунок 4):

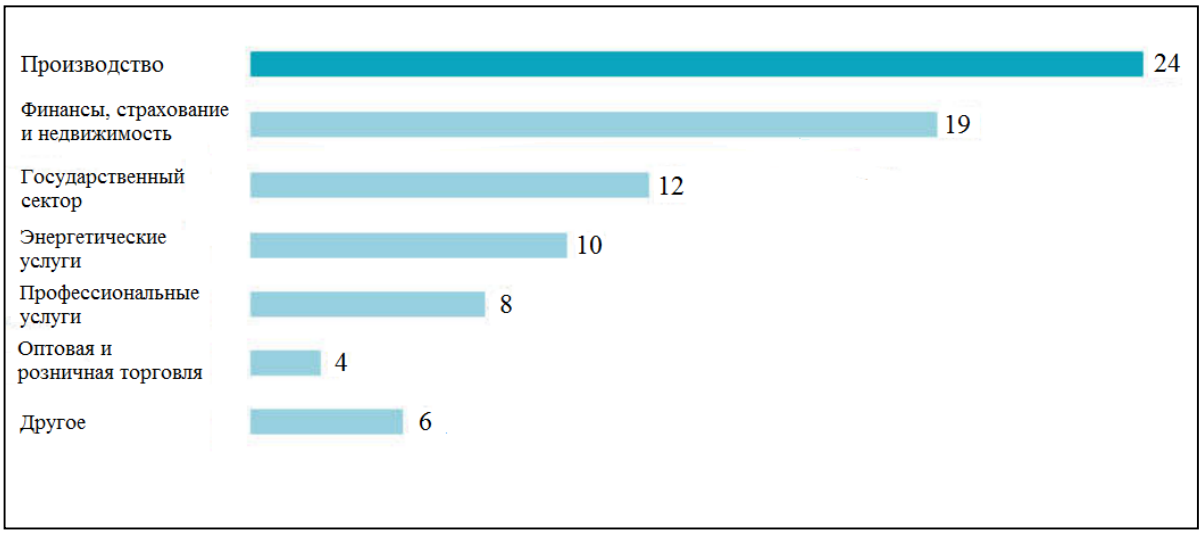

Рисунок 4 – Распределение кибератак по отраслям в процентах [8]

Учитывая быстрый темп цифровой трансформации, которая охватила весь мир, проблема информационной безопасности на производстве с каждым годом становится все более актуальной, а кибератаки все более распространенными.

Согласно отчету Verizon 2018 года (2018 Data Breach Investigations Report by Verizon), более 30% нарушений данных в производственных компаниях в 2017 году были связаны с кражей интеллектуальной собственности. Данный отчет показал, что 86% кибератак, с которыми столкнулись производственные компании, были целевыми атаками [9]. Хорошо известно, что украденная интеллектуальная собственность может использоваться для копирования продуктов и процессов, получения конкурентного преимущества и других целей.

На рисунке 5 представлены основные киберугрозы, с которыми сталкивается цифровое производство:

Рисунок 5 – Топ 10 угроз информационной безопасности для цифрового производства [10]

Для предотвращения утечек данных и прочих угроз информационной безопасности, цифровое производство должно успешно справляться как с внешними, так и с внутренними нарушителями.

Нередко бывает, что компании пренебрегают защитой от внутреннего нарушителя, основываясь на предположении о профессиональной компетенции и честности рабочего персонала, и уделяют куда больше внимания внешним угрозам.

Однако упомянутый выше отчет Verizon показал, что 46% случаев кражи интеллектуальной собственности начинаются именно с сотрудника, который сотрудничает с киберпреступниками для извлечения данных. В случае с внутренним нарушителем, доступ часто осуществляется благодаря

неправильному использованию привилегированных учетных данных. Причем внутренний нарушитель в таком случае необязательно является системным администратором: исследование Centrify [11] показало, что 52% опрошенных сотрудников IT-персонала в США предоставляли учетные данные для входа в систему подрядчика, а 59% – коллеге по работе.

Из всего этого следует крайняя непредсказуемость поведения внутреннего нарушителя, который имеет значительно больше возможностей кражи данных или несанкционированного доступа к информации, чем внешний нарушитель.

Задача обнаружения внутреннего нарушителя значительно усложняется, если злоумышленник дополнительно использует шифрование своего сетевого трафика. В таком случае традиционные методы защиты от атак (DLP, SIEM) становятся неэффективными, необходим другой способ защиты от внутреннего нарушителя.

1.4 Выводы

В условиях глобальной цифровой трансформации, компоненты производства становятся все более взаимосвязанными, образуя единое информационное пространство. Расширенная цепочка поставок становится все более тесно связанной с сетью, что позволяет обрабатывать данные о продукте на протяжении всего его жизненного цикла. Все эти решения позволяют существенно повысить качество продукта, сделать производство более автономным за счет применения автоматизированных систем, уменьшить расходы на производство.

Однако данная концепция единого цифрового пространства помимо значительного повышения эффективности производства в то же время открывает новые просторы для киберпреступников.

Вне зависимости от конечной цели атаки, киберпреступность неизбежно приносит большие убытки её жертвам, для производственных компаний это выражается в потере дохода и клиентов.

На рисунке 6 представлены финансовые потери компаний во всем мире в триллионах долларов:

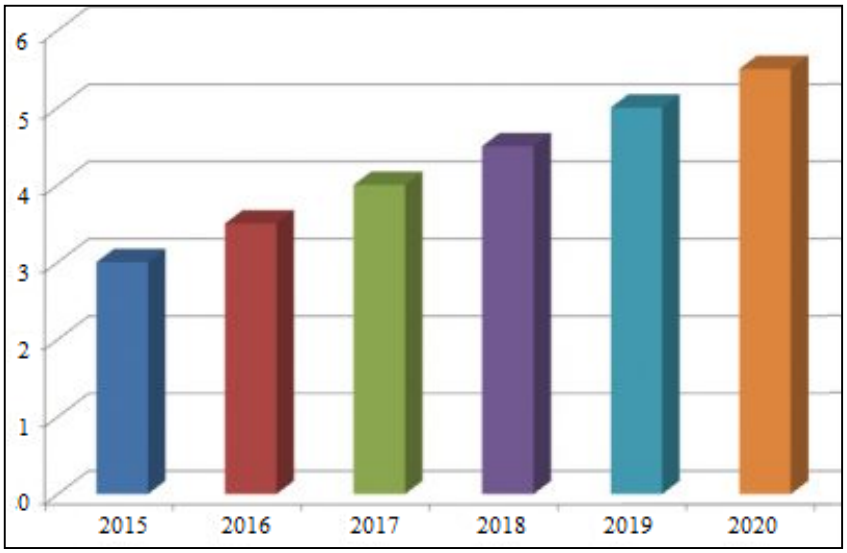

Рисунок 6 – Финансовые потери компаний от киберпреступности в триллионах долларов [12]

Официальный ежегодный отчет о киберпреступности за 2019 год, основанный на исследовании Cybersecurity Ventures [12], предсказывает, что к 2021 году киберпреступность будет стоить компаниям во всем мире 6 триллионов долларов. Этот показатель превышает показатель на 2015 год в целых два раза, что свидетельствует о крайне стремительном увеличении уровня киберпреступности.

Большое количество пользователей, компьютеров, систем и подсетей увеличивает число связей как с внешним миром, так и внутри подсистем, что делает производственный сектор чрезвычайно уязвимым для кибератак.

Для сохранения конкурентоспособности и защиты подсистем от атак, нацеленных на нарушение информационной безопасности, предприятия

применяют различные методы противодействия как внешним, так и внутренним злоумышленникам.

Системы обнаружения вторжений, защищающие от внешних атак, DLP и SIEM-системы, нацеленные на защиту от внутреннего нарушителя, являются традиционными средствами, применяемыми во всем мире. Однако с увеличением темпов цифровой трансформации и использовании в связи с этим все более совершенных и изощренных методов кибератак, злоумышленника становится все сложнее обнаружить и, соответственно, противостоять ему. Очевидно, что для решения данной проблемы необходимо использовать новые методы защиты, которые могли бы усовершенствовать существующие системы обеспечения информационной безопасности.

2 МОДЕЛЬ ТИПОВОГО ОБЪЕКТА КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

Модель Лотки — Вольтерры используется для описания динамики экологических систем, в которых взаимодействуют две популяции, а именно популяция хищников и популяция жертв. С помощью этой модели можно предсказать исход межвидовой конкуренции в зависимости от начальных параметров (размера популяции и коэффициентов). Модель описывается системой дифференциальных уравнений вида:

$$\begin{cases} \frac{dx}{dt} = (\alpha - \beta y)x \\ \frac{dy}{dt} = (\delta x - \gamma)y \end{cases} \quad (1)$$

Где:

- x и y – зависимости от времени t, которые характеризуют количества жертв и хищников соответственно*;*
- α – вероятность того, что жертвы размножатся;
- γ – вероятность того, что хищник умрет от голода;
- β – вероятность того, что жертва будет съедена хищником;
- δ – вероятность того, что хищнику хватит еды на дальнейшее размножение.

Система уравнений, представленная выше, основана на следующих допущениях:

1. В отсутствии хищников жертвы размножаются неограниченно:

$$dx = (\alpha - \beta * 0)xdt$$

$$x(t) = \int_{t_0}^{t_n} dxdt = \int_{t_0}^{t_n} \alpha x_t \, dt \implies x(t) = e^{\alpha t}$$

2. Хищники в отсутствии жертв вымирают:

$$dy = (\delta * 0 - \gamma)ydt$$

$$y(t) = \int_{t_0}^{t_n} dydt = \int_{t_0}^{t_n} -\gamma y_t \, dt \implies x(t) = e^{-\gamma t}$$

Таким образом, в системе (1) зависимость $\frac{dx}{dt}$ характеризует плотность популяции жертв (чем меньше хищников, тем больше жертв), а зависимость $\frac{dy}{dt}$ – плотность популяции хищников (чем больше жертв, тем больше хищников).

Стационарная позиция, в которой изменения численности обеих популяций равна нулю ($\frac{dx}{dt} = \frac{dy}{dt} = 0$), достигается в следующих точках:

$$\begin{cases} \bar{x} = \dfrac{\gamma}{\delta} \\ \bar{y} = \dfrac{\alpha}{\beta} \end{cases}$$

Спроецируем данную модель на промышленную КИИ, состоящую из беспроводных датчиков (IIoT), в которой каждый узел является маршрутизатором и предназначен для сбора, обработки и передачи данных операторам (отслеживание цели, мониторинг и управление, интеллектуальное обнаружение и др.). Датчики из КИИ могут находится в двух состояниях:

КИИ подвержена различным кибератакам:

- Wormhole, Blackhole, Sinkhole, Hello flood, detour attack;
- Диверсия – использование злоумышленником неправильных метрик при построении маршрутов, запрошенных обычными узлами;
- Переполнение ретрансляция служебных запросов на несуществующий адрес и др.
- Имперсонация;
- Фальсификация (отправка ложных сообщений);
- Вывод узла из строя и др.

Однако, некоторые из приведенных атак направлены не на самих участников сети (узлов – жертв), а на потребление ресурсов и несанкционированный доступ к информации. Следовательно, они не могут быть рассмотрены в контексте данной модели, поскольку такие виды атак никак не влияют на взаимодействие двух видов популяций, в отличие от атак на маршрутизацию, DoS и имперсонации.

Учитывая данный факт определим необходимые понятия для описания предлагаемой модели:

1. Атака в контексте рассматриваемой среды – это вредоносное вмешательство в работу маршрутизации КИИ, а также вмешательство в работоспособность отдельных устройств (датчиков) с целью выведения их из строя или имперсонации (узел – злоумышленника выдает себя за другой узел, чтобы воспользоваться какими-либо привилегиями имитируемого узла);
2. Устройство считается выведенным из строя, если оно лишено работоспособности (в терминах биологической модели Лотки – Вольтерры – мертвая жертва);
3. Жертвой назовём узел, который выполняет свою основную функцию, никаким образом не препятствует работе сети и не атакует остальных участников сети.
4. Инфицированный узел – это узел, который передаёт по сети не только доброкачественную полезную нагрузку, но и вредоносные пакеты.
5. Вакцинированный узел – это узел, который в период действия вакцинации является наиболее защищённым от атак в сравнении с остальными узлами.
6. Хищник – это узел, который осуществляет атаки на узлы – жертвы или на протокол маршрутизации.

Множество узлов КИИ может быть представлено в виде совокупности пяти классов (рисунок 7).

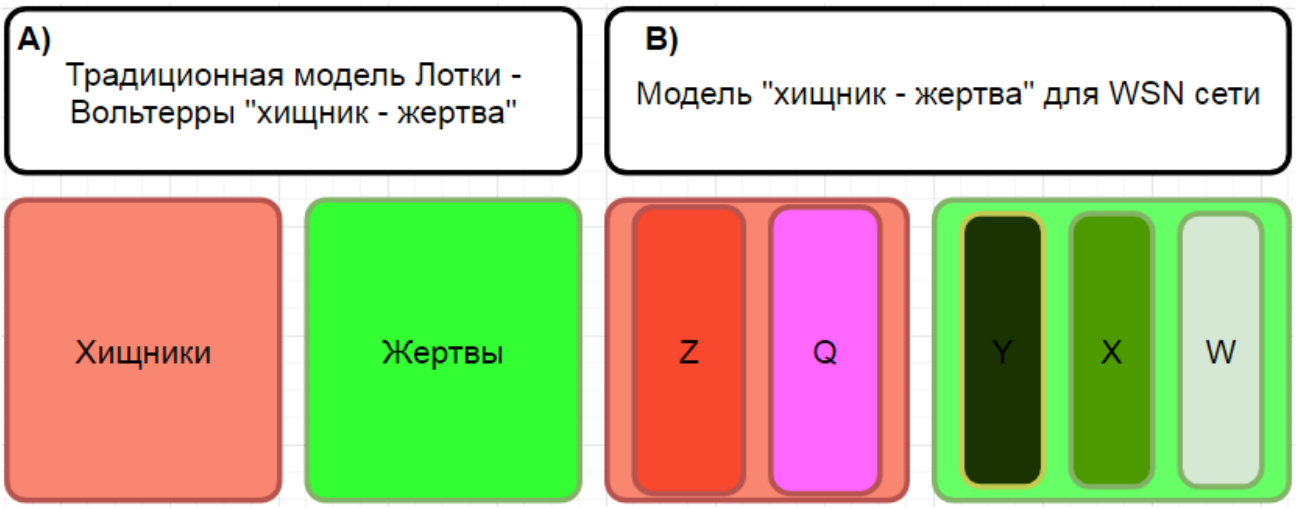

Рисунок 7 – А) традиционная модель «хищник - жертва»; В) предлагаемая модель для WSN – сети из датчиков

Для описания модели определим следующие классы:

Класс X – класс жертв. Узлы, принадлежащие данному классу, успешно выполняют целевую функцию. При этом узлы данного класса являются «вакицнированными» (на них установлено дополнительное защитное ПО).

Класс W – класс жертв. Узлы, принадлежащие данному классу, успешно выполняют целевую функцию.

Класс Y – класс жертв. Узлы, принадлежащие данному классу, выведены из строя, при этом они не представляют угрозы для других узлов и могут быть либо восстановлены в класс W, либо могут со временем перейти в состояние D (рис. 3);

Класс Z – класс хищников. Узлы, принадлежащие данному классу, находятся под прямым управлением злоумышленников, реализуют кибератаки и стремятся перевести узлы из классов X и W в классы Q и Y в зависимости от целей и мотивов.

Класс Q – класс хищников. Узлы, принадлежащие данному классу, являются «узлами-зомби». Они наравне с Z «охотятся» на узлы из X и W,

стремясь перевести их в класс Y, и так же, как и Y могут быть восстановлены в класс в W.

Классы предлагаемой модели образуют трёхуровневую иерархию, изображенную на рисунок 8.

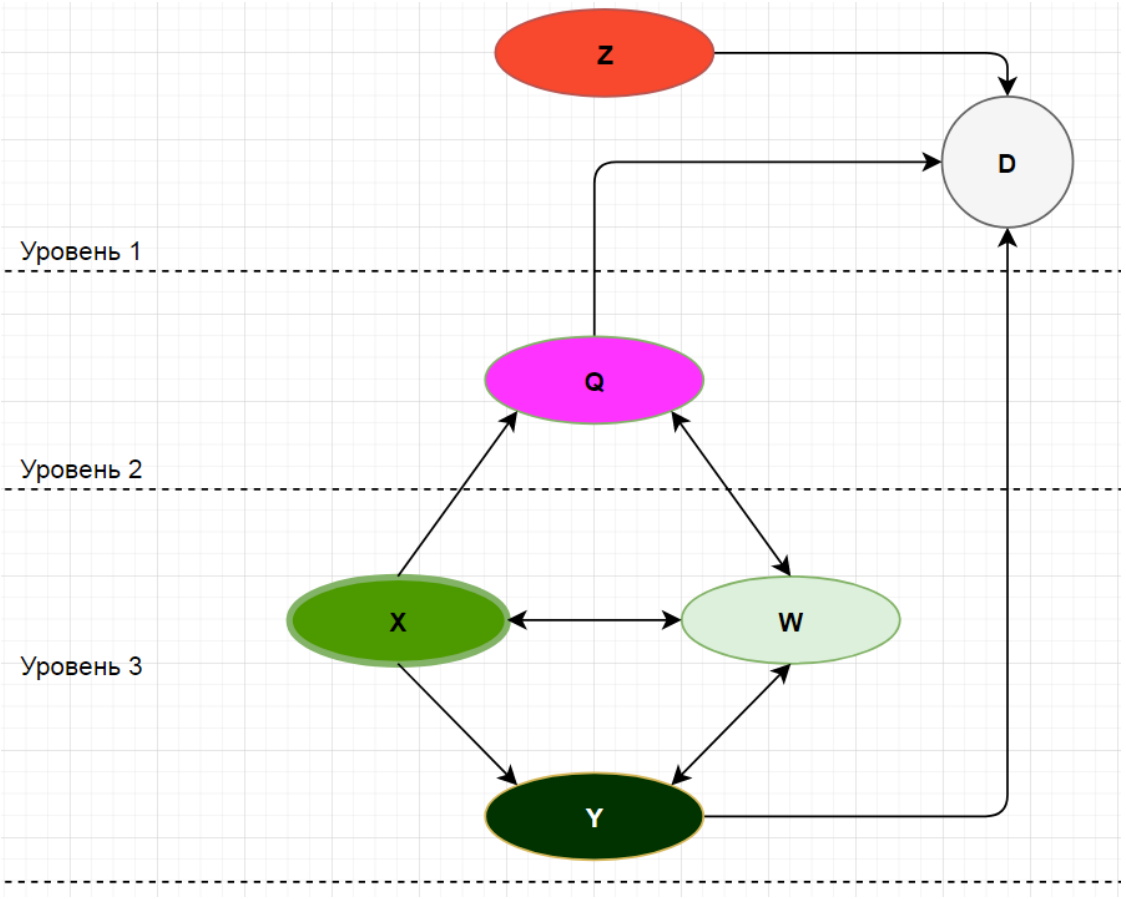

Рисунок 8 – Иерархия классов

Стрелки между классами показывают диапазон состояний каждого класса (Например, узел из Q может перейти в класс W, а потом может быть вакцинирован и присоединится к классу X и т.д.).

Состояние D – это терминальное состояние, переход из которого невозможен. Оно описывает выход узлов из строя. Узлы Y переходят в это состояние, если их не успели восстановить в класс W, а узлы из Q и Z становятся мертвыми (переходят в D) при попадании в них функции охотника.

Та же иерархия с весовыми коэффициентами представлена на рисунке 9.

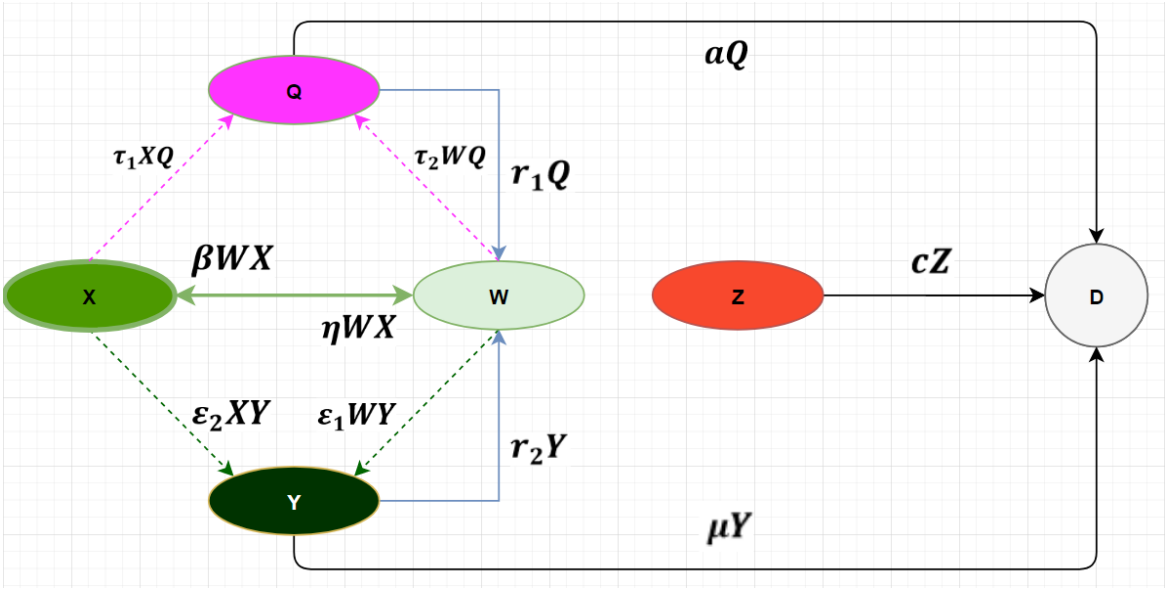

Рисунок 9 – Иллюстрация модели «хищник-жертва»

Введём следующие коэффициенты (таб. 1):

- Веса r_1, r_2 характеризуют вероятности, с которыми могут быть «вылечены» узлы из Q и Y соответственно;
- Узлы класса Z инфицируют узлы X и W с вероятностями τ_1 и τ_2 соответственно
- Под воздействием узлов из Q и Z жертвы W и X могут перейти в класс Y с коэффициентами ε_1 и ε_2 соответственно;
- μ, a, c – коэффициенты вымирания узлов из класса Y, Q, Z.

Существует также переход между классами X и W:

- Узлы W становятся вакцинированными (переходят в X) с вероятностью β;
- Устойчивость к заражению (вакцинация) пропадает со скоростью прямо пропорциональной коэффициенту η.

Таблица 1 – Коэффициенты модели

	W	X	Y	Q	Z	D

W		β	ε_1	τ_2		
X	η		ε_2	τ_1		
Y	r_2					μ
Q	r_1					a
Z						c
D						

Класс X является «вакцинированным», поэтому он менее восприимчив к атакам в сравнении с классом W. Следовательно, коэффициент заражения узлов X классом Y должен быть выше: $\varepsilon_1 \geq \varepsilon_2$. Относительно класса Q коэффициенты τ_1 и τ_2 находятся в той же зависимости: $\tau_2 \geq \tau_1$.

В отличие от узлов класса Y, классы Q, Z могут вымирать только под воздействием внешних факторов. Таким фактором может быть некая дополнительная зависимость, которая будет влиять на значение коэффициентов a и c. Обозначим эту зависимость как $h_\Phi(t)$ и назовём «охотником».

По мимо зависимости от времени, функция $h_\Phi(t)$ имеет динамический параметр $\Phi = \{\varphi_1, \dots, \varphi_n\}$, который представляет собой исследуемый в данный момент времени t набор характеристик пересылаемых пакетов в WSN – сети (рис 1).

На основе входных данных $\{\Phi_0, t_0\}$ функция $g_\Phi(t)$ определяет, какой из дальнейших метод анализа сетевых данных должен быть выбран для наиболее точной классификации между атакой и нормальным поведением узла в сети. Такие методы анализа основаны на известных алгоритмах выравнивания последовательностей:

- Нидлман – Вунш (глобальное выравнивание);
- Смит – Вотерман (локальное выравнивание);
- Алгоритм выравнивания «Gotoh».

Сформулируем систему уравнений предлагаемой расширенной модели «хищник-жертва»:

$$\begin{cases} \dfrac{dW}{dt} = w_g W\left(1 - \dfrac{W}{N_w}\right) + WX(\eta - \beta) - \varepsilon_1 WY - \tau_2 WQ + r_2 Y + r_1 Q - m_W W \\ \dfrac{dX}{dt} = x_g X\left(1 - \dfrac{X}{N_X}\right) + WX(\beta - \eta) - \varepsilon_2 XY - \tau_1 XQ - m_X X \\ \dfrac{dY}{dt} = y_g Y\left(1 - \dfrac{Y}{N_Y}\right) + \varepsilon_1 WY + \varepsilon_2 XY - r_2 Y - Y(\mu + m_Y) \\ \dfrac{dQ}{dt} = q_g Q\left(1 - \dfrac{Q}{N_Q}\right) + \tau_2 WQ + \tau_1 XQ - r_1 Q - aQ \\ \dfrac{dZ}{dt} = z_g Z\left(1 - \dfrac{Z}{N_Z}\right) - cZ \end{cases} \quad (1)$$

Где:

- t – время;
- N_w, N_X, N_Y, N_Q, N_Z – количество узлов, принадлежащих классу W, X, Y, Q, Z соответственно;
- w_g, x_g, y_g, q_g, z_g – коэффициенты, характеризующие рост количества узлов в классах W, X, Y, Q, Z соответственно;
- m_X, m_Y, m_W – вероятность промаха охотника (попадания по узлам из классов X, Y, W).

Таким образом, задача охотника заключается в том, чтобы максимизировать коэффициенты вымирания для инфицированного класса Q и хищника Z, при этом не влияя на остальные классы. Другими словами, должно поддерживаться состояние системы, при котором вероятности промаха должны стремится к нулю (2), а коэффициенты вымирания Q и Z – к своему максимуму. Учитывая, что узлы Q могут быть восстановлены в класс W, то относительно класса Q появляется также коэффициент r_1, который наравне с a, может быть устремлен к своему максимуму:

$$\begin{cases} \begin{cases} m_W \to 0 \\ m_X \to 0 \\ m_Y \to 0 \end{cases} \\ \begin{bmatrix} r_1 \to \infty \\ a \to \infty \end{bmatrix} \\ c \to \infty \end{cases} \qquad (2)$$

3 МЕТОД ВЫБОРА ЗАЩИТНЫХ МЕХАНИЗМОВ

Нейро-нечеткие системы используются для моделирования работы трех блоков: блока фаззификации, блока вывода и блока дефаззификации.

Блок фаззификации занимается вычислением степени принадлежности четких входных параметров (точных значений, например, стоимость актива равна 2 тыс. долларов) нечетким подмножествам (например, «средний уровень угрозы»).

Блок вывода представляет собой набор логических правил, которые задают причинно-следственные отношения между входными и выходными величинами (как правило используются логические операторы «И», «ИЛИ»; например, if [стоимость актива = 2 тыс. долларов] И [вероятность наступления события = 0.7], то риск средний).

Блок дефаззификации занимается вычислением четкого выходного значения на основе результирующей функции принадлежности, которая рассчитывается механизмом вывода в блоке вывода.

Структура нейро-нечеткой системы представлена на Рисунке 10.

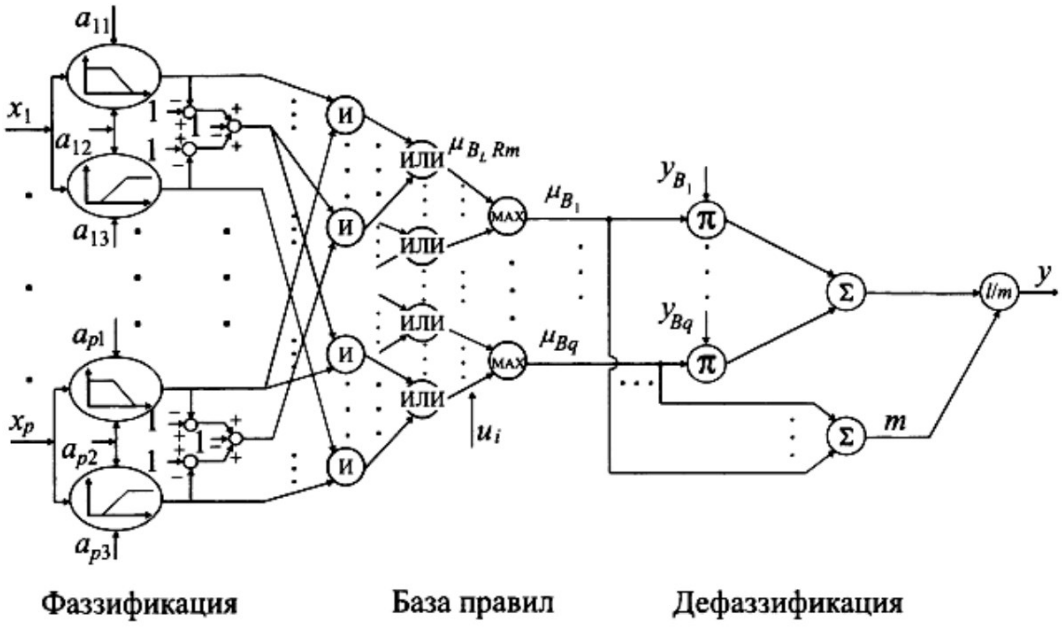

Рисунок 10 – Структура нейро-нечеткой системы [1]

Модель отображает входной вектор X в выходной вектор Y, обеспечивая как можно более точную аппроксимацию реальной системы. Это отображение может быть представлено декартовым произведением X×Y, которое является геометрической поверхностью отображения. Каждому логическому правилу из базы правил соответствует нечеткая точка в этом пространстве [2].

Настройка нейро-нечеткой модели производится на основе обучающей выборки, которая содержит значения одного или нескольких входных параметров, и значение выходного параметра моделируемой системы с использованием метода обратного распространения ошибки или другого применяемого для нейронных сетей метода [3].

Таким образом, нейро-нечеткая модель позволяет [4]:

– оптимизировать и настраивать параметры функции принадлежности на основе измерений входных и выходных

зависимостей реальной системы;

– обеспечить возможность корректировки нечетких моделей в случае, если они сформированы недостаточно точно (например, в результате экспертной оценки);

– расширять формируемые экспертами нечеткие модели на те области системы, знания о которых у экспертов ограничены.

Построение нейро-нечеткой модели было проведено на основе системы нейро-нечеткого вывода ANFIS (adaptive neuro-fuzzy inference system) в программном средстве MATLAB. Был использован специализированный пакет Neuro-Fuzzy Designer [5].

Интерфейс взаимодействия выглядит следующим образом (Рисунок 11):

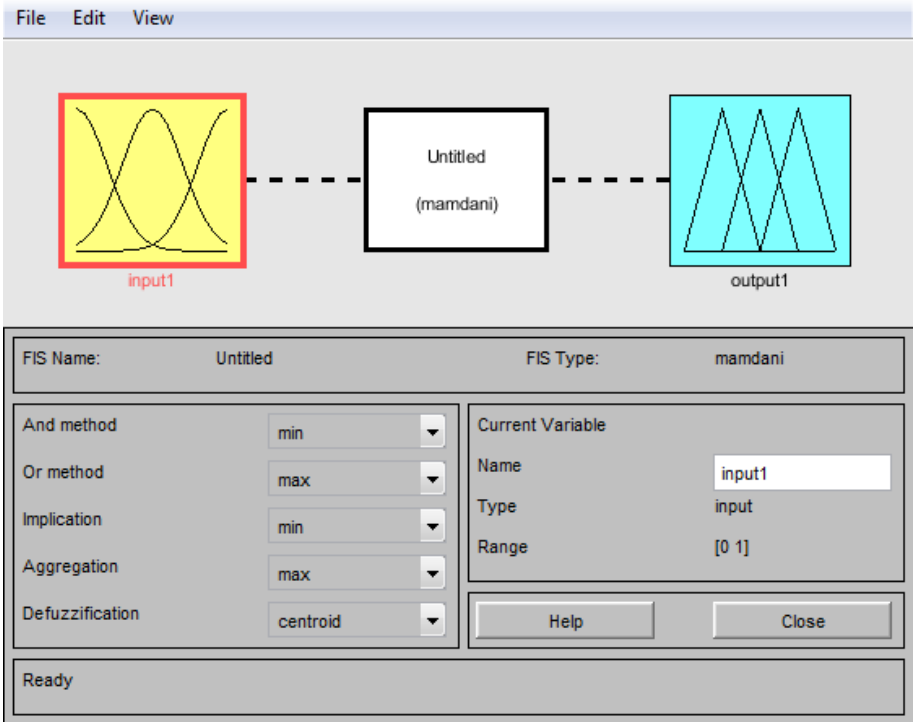

Рисунок 11 – Интерфейс взаимодействия

Как видно из рисунка, система представляет собой блок входов, блок нечеткой модели и блок выходов. Имеется возможность

отредактировать функции принадлежности, количество входов, саму нечеткую модель (на рисунке это mamdani).

Перед тем как задать в качестве нечеткой модели нейро-нечеткую модель, необходимо определиться непосредственно с задачей: какие входные данные нужно использовать, что необходимо получить на выходе.

Смоделируем сценарий, при котором была проведена экспертная оценка рисков информационной безопасности по следующим критериям:

- вероятность наступления события (значение от 0 до 1);
- серьезность угрозы (по 10-балльной шкале);
- стоимость актива (в тыс. долларах).

На основе значений этих критериев экспертом устанавливается значение риска (в диапазоне от 1 до 3, где 3 – самый высокий, а 1 – самый низкий).

Сведения об экспертной оценке представлены в Таблице 1.

Таблица 1- Результаты экспертной оценки

Вероятность наступления события (от 0 до 1)	**Серьезность угрозы (от 0 до 10)**	**Стоимость актива (в тыс. долларов)**	**Значение риска (от 1 до 3)**
0.1	10	2	3
0.6	6	4	3
0.2	8	3	2
0.7	4	3	2
0.8	1	1	1
0.1	2	5	1

Для импортирования этих данных использован Workspace в интерфейсе MATLAB (Рисунок 12):

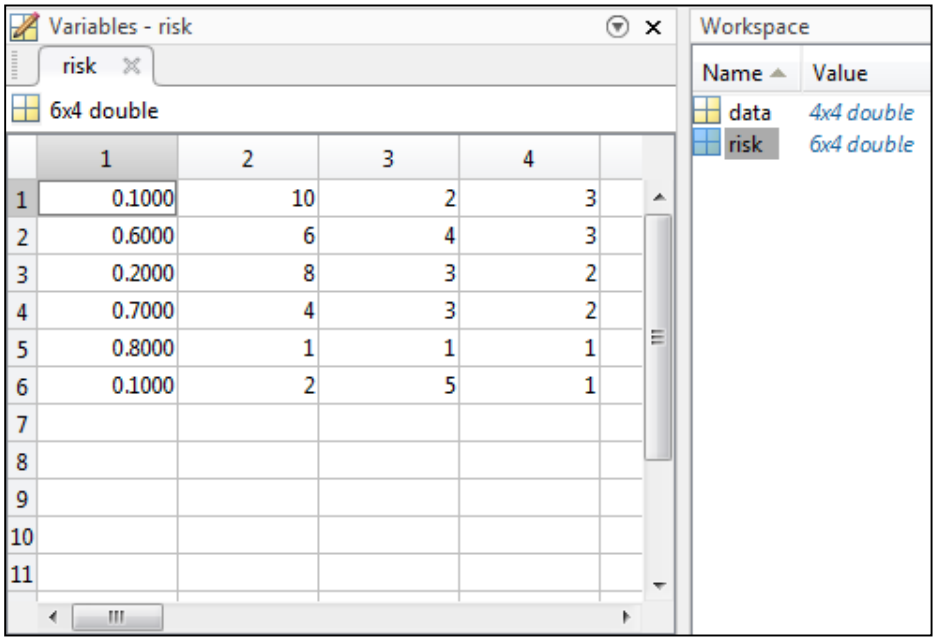

Рисунок 12 – Входные данные

На основе этих входных данных построена следующая нейро-нечеткая система (Рисунок 13):

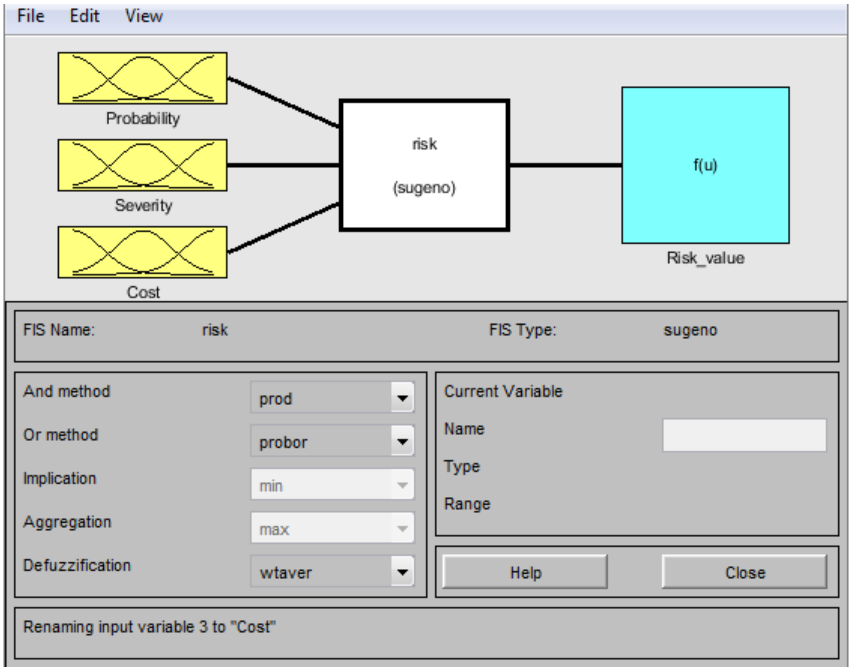

Рисунок 13 – Логическое представление нейро-нечеткой системы

Как видно из Рисунка 13, теперь на вход нечеткой модели поступает 3 отдельных входа: вероятность наступления события (Probability), серьезность события (Severity) и стоимость актива (Cost). На выходе будет единственное значение риска.

Также имеется структурное представление созданной нейро-нечеткой системы (Рисунок 14):

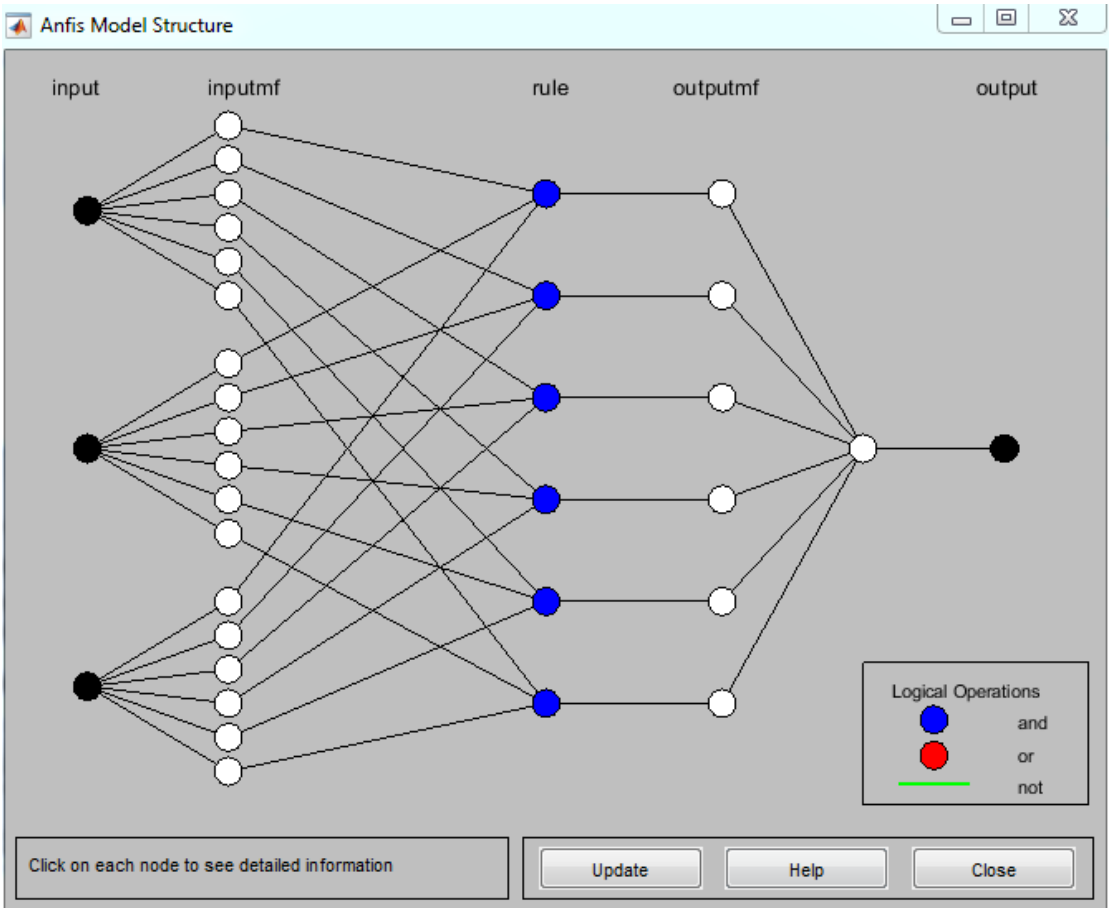

Рисунок 14 – Структура сгенерированной нейро-нечеткой системы

Проведем несколько экспериментов, чтобы убедиться в корректности построенной модели.

Интерфейс Rule Viewer позволяет внести произвольные входные данные для построенной модели и получить выходное значение на

основе этих данных. Для начала проверим значения из таблицы: те значения, которые изначально подавались при создании модели (Рисунок 15):

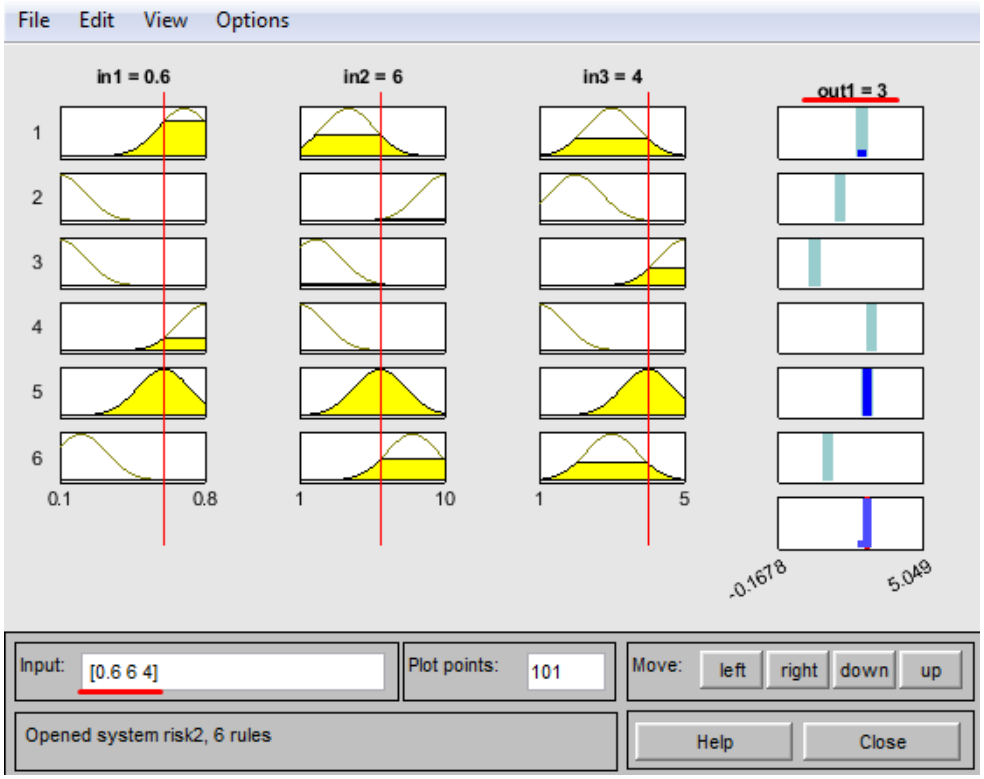

Рисунок 15 – Проверка значений [0.6 6 4]

По входному набору [0.6 6 4] (вероятность, серьезность, цена актива) получено значение риска 3. Значение ожидаемо совпадает с таблицей.

Теперь введем данные, которых изначально не было в таблице: промежуточные данные, то есть те данные, которые по той или иной причине эксперт не получил во время оценки (Рисунок 16).

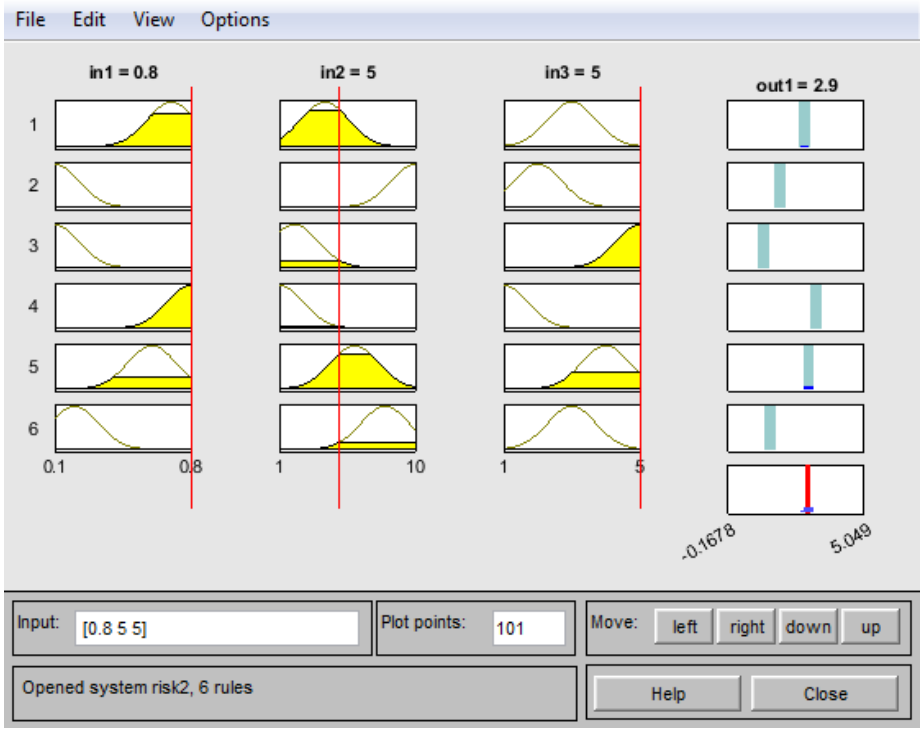

Рисунок 16 – Проверка значений [0.8 5 5]

Несмотря на то, что значение серьезности и не является настолько высоким (5 из 10), высокая вероятность наступления события (0.8) и стоимость актива (5) приводит к тому, что значение риска составляет 2.9, что может быть отнесено к значению «высокий риск». То есть, несмотря на то, что подобного набора данных не подавалось на обучение модели, модель смогла предоставить логичный и точный результат.

Рассмотрим другой граничный случай (Рисунок 17):

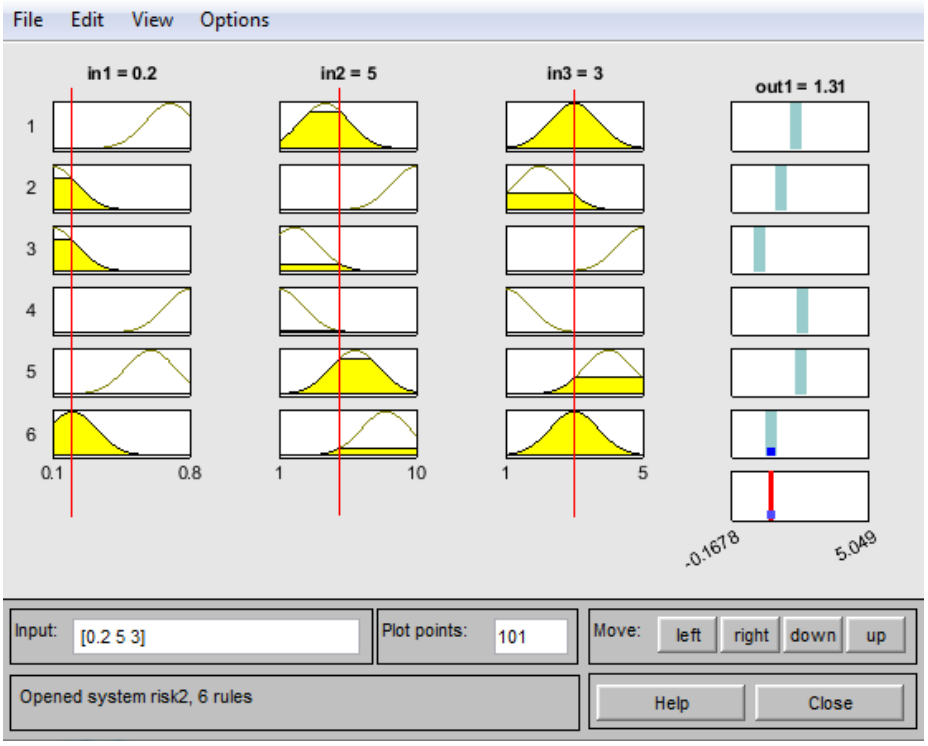

Рисунок 17 – Проверка значений [0.2 5 3]

В этом примере рассматривается такое же значение серьезности риска (5 из 10), однако низкая вероятность наступления (0.2) и не самая высокая стоимость актива (3) приводят к тому, что конечный риск принимает значение 1.31, что будет отнесено ближе к значению «низкий риск».

Эксперименты с небольшим отклонением значений от табличных также показали корректный и логичный результат оценки риска.

Также стоит отметить следующее удобство реализованной нейро-нечеткой системы: выход модели представляет собой точное значение риска в виде вещественного числа. Это позволяет интерпретировать результат с разной степенью детализации в зависимости от поставленной задачи.

Не важно, на сколько интервалов разбит промежуток от 1 до 3, это может быть как два интервала вида «приемлемый»/«неприемлемый» риск, так и более мелкие разбиения по типу «средний», «выше среднего» и так далее. Оценка не будет «огрублена» из-за нечетких множеств, нейро-нечеткая модель даст точную оценку.

4 МЕТОД ВЫБОРА РЕСУРСОВ ПЛАТФОРМЫ ЭЛАСТИЧНЫХ ВЫЧИСЛЕНИЙ

Программно-определяемая сеть (англ. software-defined networking, SDN; также программно-конфигурируемая сеть) — сеть передачи данных, в которой уровень управления сетью отделён от устройств передачи данных и реализуется программно, одна из форм виртуализации сети. Эластичность ПКС делает этот подход привлекательным для удовлетворения требований к построению эффективной системы управления безопасностью КИИ. Применение принципов ПКС обеспечивает программируемость и гибкость, а также позволяет добиться одновременного упрощения управления сетью и предоставления новых (например, V2V и V2I).

4.1 Архитектура платформы

Были разработаны 3 варианта архитектуры сетей КИИ с использованием технологии SDN:

1) Архитектура с центральным звеном управления - серверами, решающими задачи безопасности (рисунок 18):

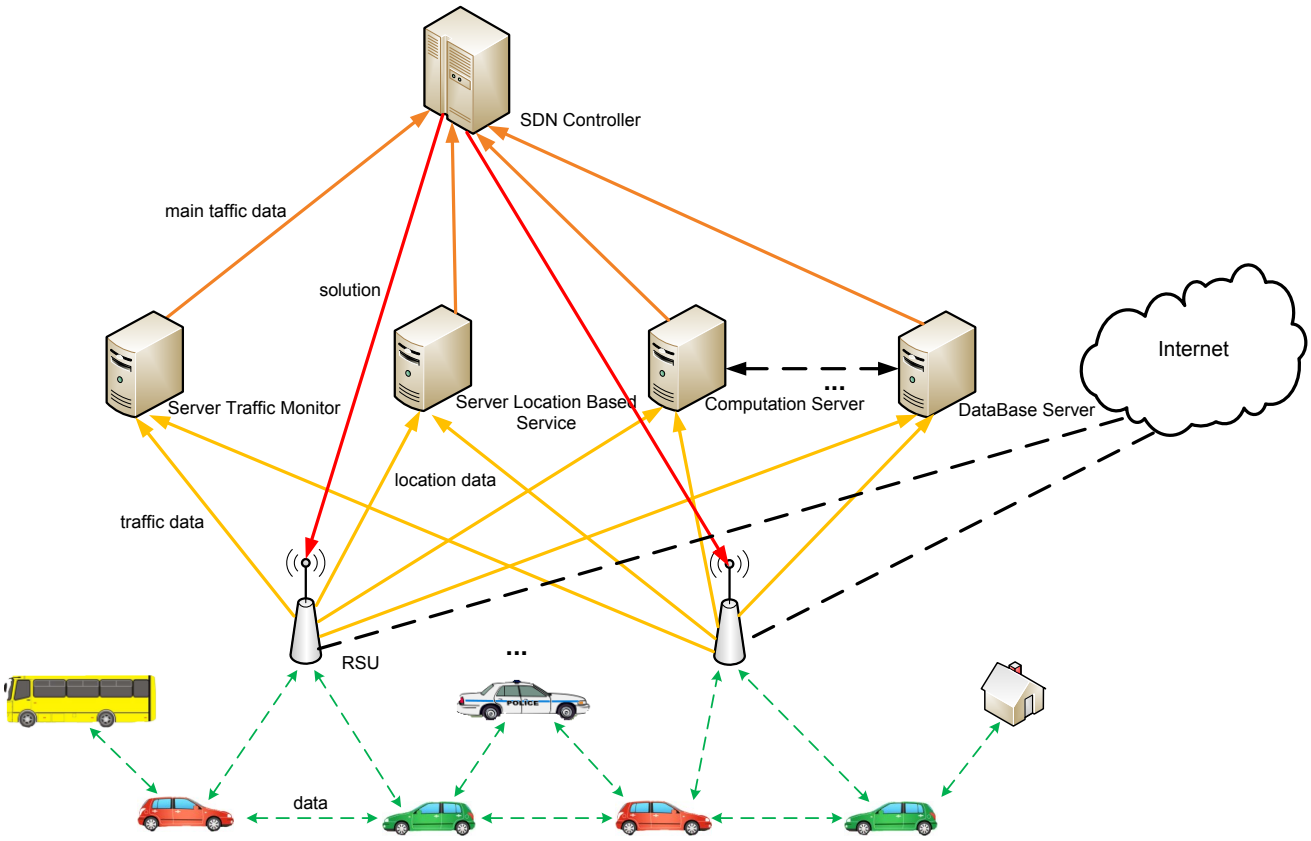

Рисунок 18 – Архитектура с центральным звеном управления и серверами

В состав архитектуры входят:

- программно-конфигурируемый контроллер (SDN Controller): логический центральный интеллект SDN, управляющий поведением сети всей системы VANET. Контроллер получает обработанную серверами информацию, и на основе полученных данных устанавливает необходимые правила, определяет параметры маршрутизации, принимает решения об увеличении/уменьшении мощности передатчиков конкретных узлов в сети и т.д.;
- сервера: устройства, выполняющие сбор определенной информации о сети, производящие анализ полученной информации. Результаты анализа обрабатываются, после чего отправляются на программно-конфигурируемый контроллер. На серверах могут находиться различные анализаторы трафика, сервисы, отвечающие за обработку геолокации конечных пользователей, базы данных, содержащие информацию о дорожной обстановке, автомобилях и их владельцах. Кроме того

предусмотрен сервер вычислений, осуществляющий расчёт скорости автомобиля, дистанции, оценка дорожной обстановки. При данной архитектуре VANET сети сервера могут быть связаны между собой. Например, сервер вычислений связан с базой данных, которая постоянно обновляется, и использует хранящиеся в базе данных знания для осуществления последующих вычислений;

- придорожный сетевой узел (RSU): развернутые вдоль дороги устройства, связывающие конечных пользователей, участников дорожного движения, с серверами и программно-конфигурируемым контроллером, также обеспечивая доступ узлов в интернет. ПСУ могут являться маршрутизатором, мостом, точкой доступа и клиентом;
- конечные пользователи VANET сети, участники дорожного движения и объекты инфраструктуры города: автомобили, умные дома, ДПС.

2) Архитектура с частичной децентрализацией (рисунок 19):

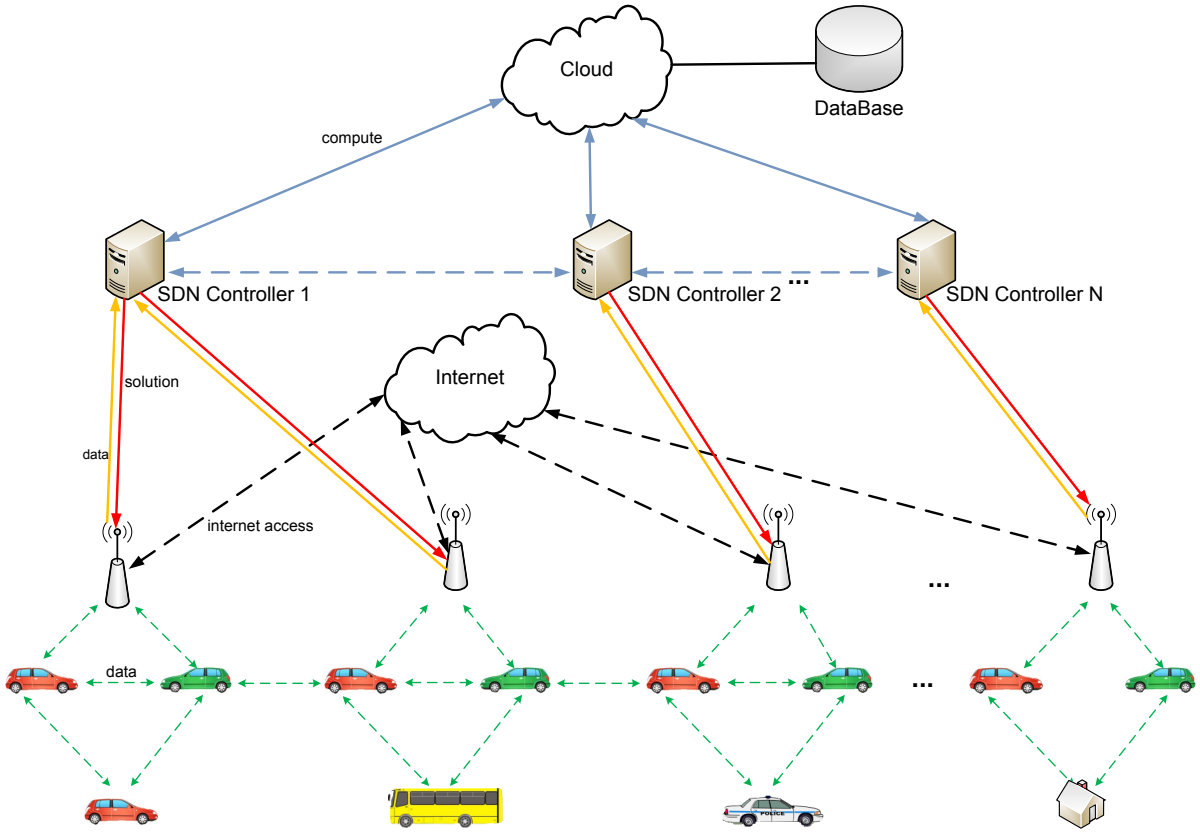

Рисунок 19 – Архитектура с частичной децентрализацией

В состав архитектуры входят:

- облако: происходят различные вычисления такие, как расчёт скорости автомобиля, дистанции, оценка дорожной обстановки. Облако вычислений связано с базой данных, которая постоянно обновляется, и использует хранящиеся в базе данных знания для осуществления вычислений;
- база данных: база, в которой хранится информация о дорожной обстановке, автомобилях и их владельцах;
- программно-конфигурируемый контроллер (SDN Controller): логический центральный интеллект SDN, управляющий поведением отдельной части сети VANET. Каждый контроллер выполняет сбор информации сети, выполняет анализ полученных данных. Контроллеры могут обмениваться между собой полученными данными, а для произведения вычислений воспользоваться облаком. На основе обработанной информации контроллеры устанавливает необходимые правила политики безопасности, определяет параметры маршрутизации в рамках своей зоны ответственности в сети VANET;
- придорожный сетевой узел (RSU): развернутые вдоль дороги устройства, связывающие конечных пользователей, участников дорожного движения, с программно-конфигурируемыми контроллерами, также обеспечивая доступ узлов в интернет. ПСУ могут являться маршрутизатором, мостом, точкой доступа и клиентом;
- конечные пользователи VANET сети, участники дорожного движения и объекты инфраструктуры города: автомобили, умные дома, ДПС.

3) Иерархическая архитектура (рисунок 20):

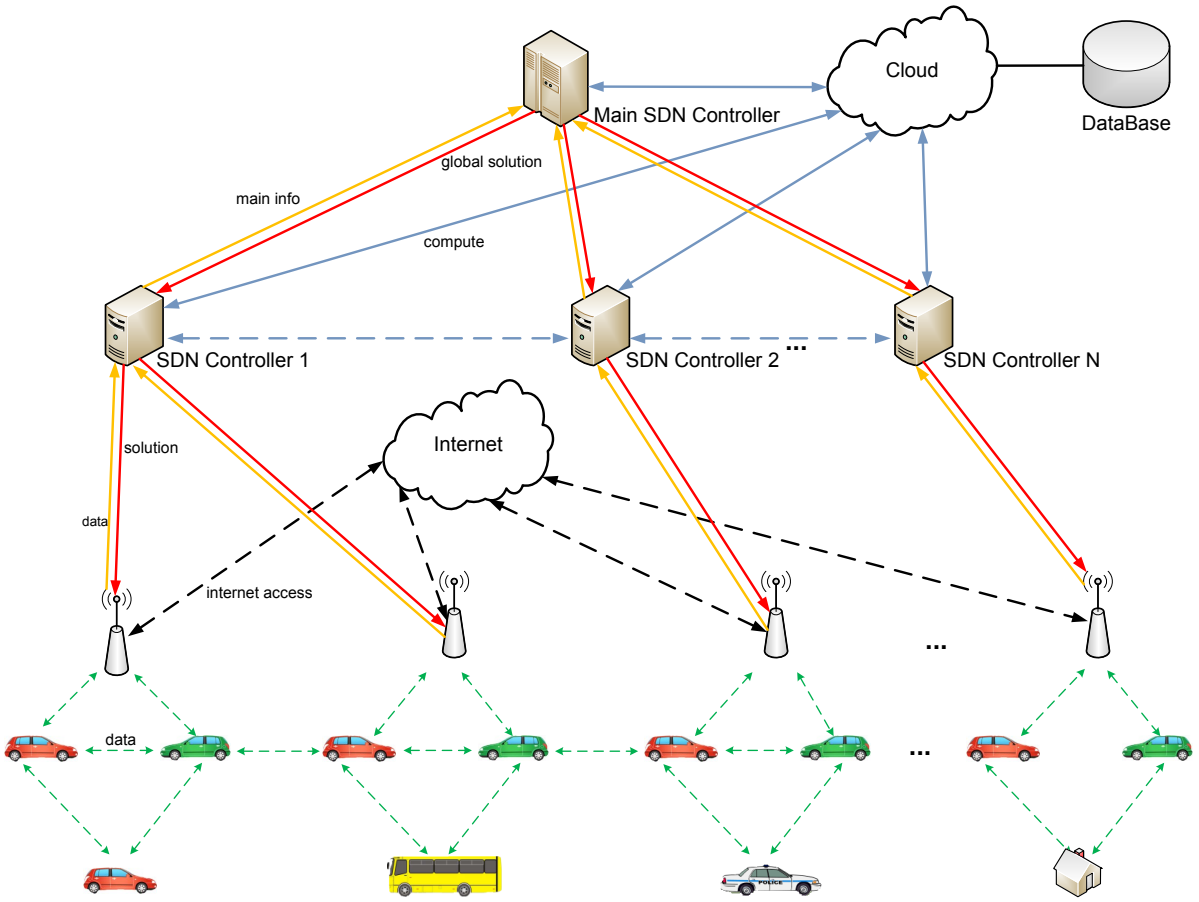

Рисунок 20 – Иерархическая архитектура

В состав архитектуры входят:

- центральный программно-конфигурируемый контроллер (Main SDN Controller): центральный контроллер рассылает правила политики для контроллеров, осуществляет деление VANET сети на зоны ответственности. Таким образом, центральный контроллер SDN имеет полное представление о состоянии всей сети VANET, что позволяет ему инструктировать контроллеры, например, о запуске конкретного протокола маршрутизации с определенными параметрами;
- облако: происходят различные вычисления такие, как расчёт скорости автомобиля, дистанции, оценка дорожной обстановки. Облако вычислений связано с базой данных, которая постоянно обновляется, и использует хранящиеся в базе данных знания для осуществления вычислений;
- база данных: база, в которой хранится информация о дорожной обстановке, автомобилях и их владельцах;

- программно-конфигурируемый контроллер (SDN Controller): логический интеллект SDN, управляющий поведением отдельной части сети VANET. Каждый контроллер выполняет сбор информации сети, выполняет анализ полученных данных. Контроллеры могут обмениваться между собой полученными данными, а для произведения вычислений воспользоваться облаком. На основе обработанной информации контроллеры устанавливает необходимые правила политики безопасности, определяет параметры маршрутизации в рамках своей зоны ответственности в сети VANET;
- придорожный сетевой узел (ПСУ): развернутые вдоль дороги устройства, связывающие конечных пользователей, участников дорожного движения, с программно-конфигурируемыми контроллерами, также обеспечивая доступ узлов в интернет. ПСУ могут являться маршрутизатором, мостом, точкой доступа и клиентом;
- конечные пользователи VANET сети, участники дорожного движения и объекты инфраструктуры города: автомобили, умные дома, ДПС.

Ниже приведена сравнительная таблица 6 предложенных вариантов архитектуры:

Таблица 6 – Сравнение VANET архитектур

	1. Классическая архитектура	2. Архитектура с центральным звеном управления - серверами, решающими задачи безопасности	3. Архитектура с частичной децентрализацией	4. Иерархическая архитектура
город 50 узлов	-/+	+	+/-	+/-
город 1000 узлов	-	-	-/+	+
шоссе 50 узлов	+/-	+	+	+/-
шоссе 1000 узлов	-	-/+	+/-	+
безопасность	-	-/+	+/-	+
маршрутизация	-	+/-	+/-	+
преимущества	простота развертывания	разделение функционала по разным серверам	равномерное распределение	гибкость управления

			нагрузки среди контроллеров	
недостатки	неэффективная маршрутизация, проблема обеспечения безопасности	большое количество узлов приводит к перегрузке контроллера	проблема определения «зон ответственности» каждого контроллера	центральный контроллер становится главным объектом атак

Можно сделать вывод, что все предложенные архитектуры превосходят по эффективности классическую VANET архитектуру. Необходимо также отметить, что эффективность архитектуры зависит от сценария, при котором она будет использоваться. Однако с учетом плотности населения и количества транспортных средств в мегаполисах наиболее целесообразно использовать иерархическую архитектуру благодаря наличию центрального контроллера, который обеспечивает распределение зон влияния контроллеров и дает возможность установления политики безопасности.

4.2 Протокол OpenFlow

Протокол OpenFlow появился на свет в 2008 году, в результате коллективной научной работы профессоров и их научных команд из разных университетов в США. Рождение протокола происходило практически одновременно с рождением парадигмы SDN.

При описании принципов работы OpenFlow принято исходить из того, что все современные коммутаторы используют в работе таблицы коммутации, построенные на основании CAM-таблиц, для передачи пакетов со скоростью среды. Этот же принцип используется при построении таблиц для большинства других сетевых технологий, таких как NAT, QoS, различных межсетевых экранов и т.д.

Работа коммутатора OpenFlow целиком основана на описанном выше принципе. Использование протокола OpenFlow дает возможность программировать таблицы потоков (коммутации) на большом количестве разных устройств. OpenFlow реализует концепцию разделения плоскостей

управления передачей данных (OpenFlow-контроллер) и непосредственно самой передачи (OpenFlow-коммутатор)[2].

На рисунке 21 представлен набор аппаратных и программных средств, которые поддерживают или могут участвовать в построении SDN сети с использованием OpenFlow. Главные роли отводятся коммутаторам и контроллерам OpenFlow. В помощь разработчикам и исследователям также доступны уже написанные приложения, реализующие тот или иной функционал сети, и средства для их отладки и мониторинга.

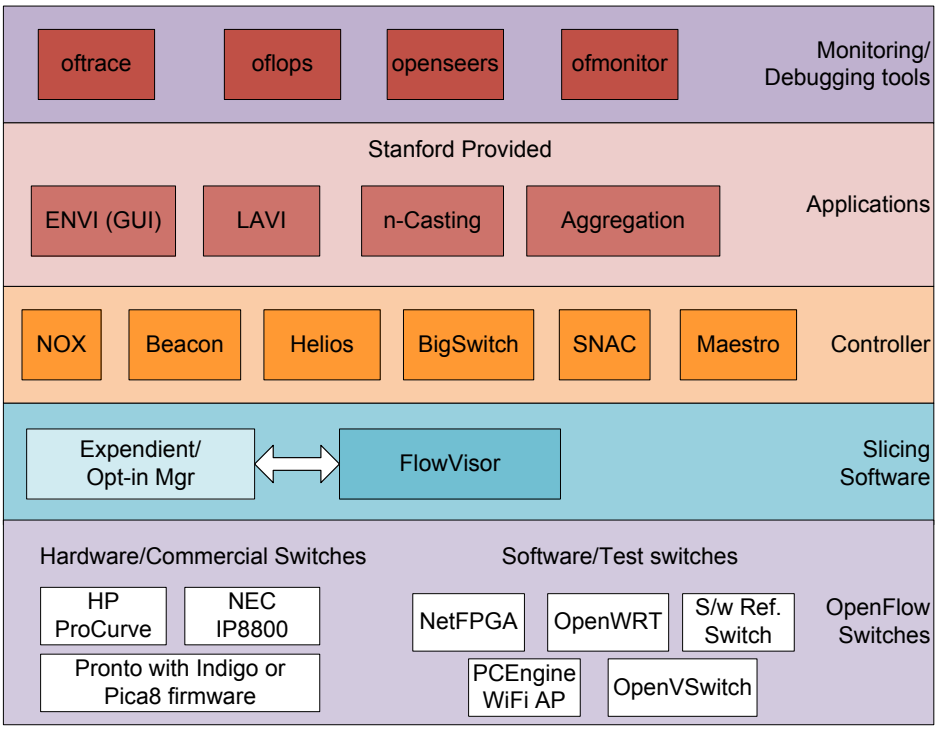

Рисунок 21– Набор аппаратных и программных средств SDN и OpenFlow

Основными компонентами программно-конфигурируемых сетей на базе протокола OpenFlow являются (рисунок 22):

- OpenFlow коммутатор;
- контроллер;
- защищенный канал, посредством которого осуществляется взаимодействие контроллера и коммутатора. Как правило, для защиты передаваемых сообщений используется TLS, однако, возможна передача по стандартному TCP без шифрования.

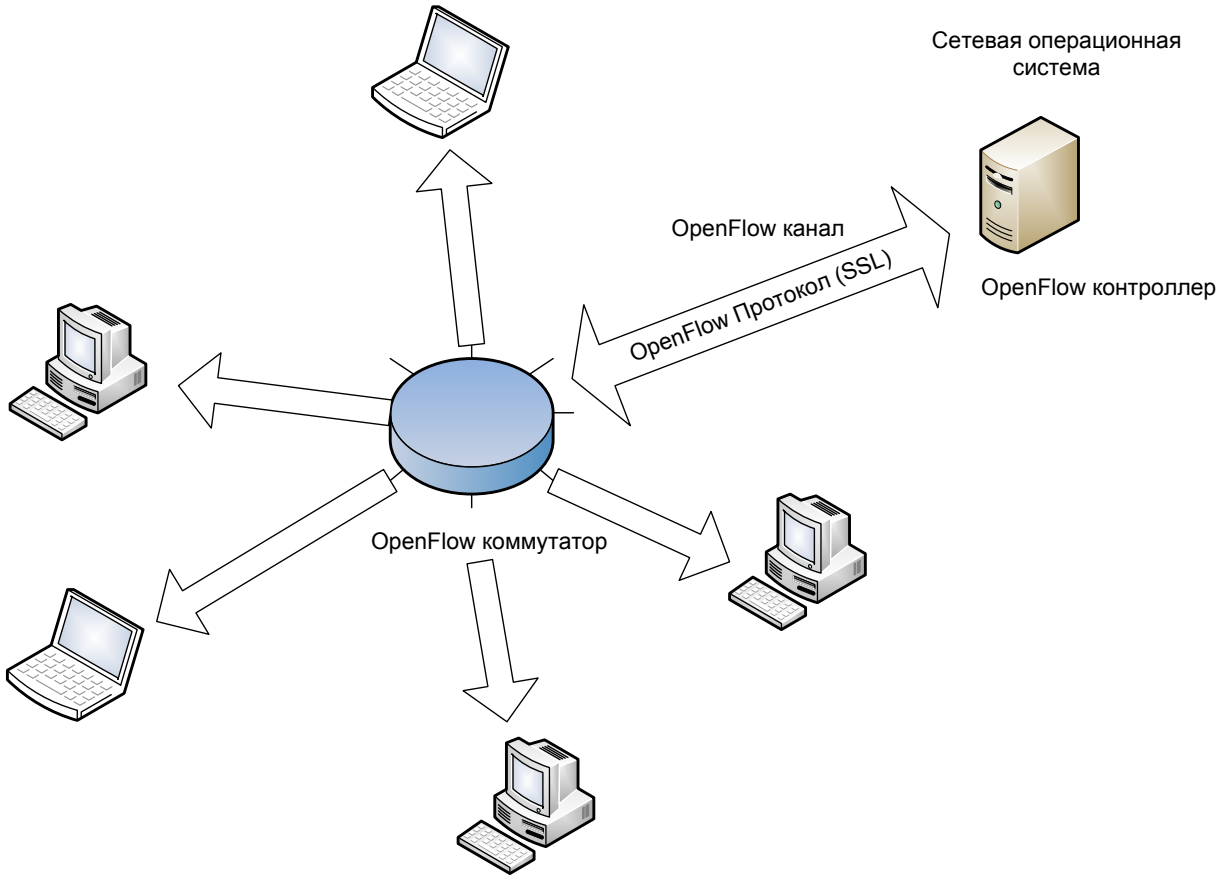

Рисунок 22 – Схема SDN сети на базе OpenFlow

Сетевая операционная система (или контроллер) в рамках концепции OpenFlow является основным, центральным звеном программно-конфигурируемых сетей (ПКС), в котором сосредотачивается вся функциональность для управления ПКС сетями. Операционная система не сама управляет сетью, а только обеспечивает программный интерфейс (API) для управления ею. Таким образом, фактически решение задач управления сетью выполняется с помощью приложений, реализованных на основе API сетевой операционной системы. Нужно отметить, что программный интерфейс должен быть достаточно общим для того, чтобы поддерживать достаточно широкий спектр приложений для решения задач управления сетью. В отличие от традиционного толкования термина СОС (Сетевая Операционная Система) как операционной системы интегрированной со стеком сетевых протоколов, в данном случае под СОС понимается программная система, обеспечивающая мониторинг, доступ, управление, ресурсами всей сети, а не конкретного узла.

СОС формирует данные о состоянии всех ресурсов сети и обеспечивает доступ к ним для приложений управления.

4.3 Архитектура платформы с использованием протокола OpenFlow

Основными элементами автомобильной сети VANET, построенной на основе программно-конфигурируемой сети, с использованием облачных технологий являются:

- контроллер SDN: логический центральный интеллект SDN, управляющий поведением сети всей системы VANET. С помощью контроллера устанавливаются нужные правила, происходит слежение за состоянием устройств, осуществляется мониторинг трафика и сбор статистики;
- коммутатор SDN: находится под управлением контроллера SDN. В задачу коммутатора входит без задержек передавать пакеты с одного порта на другой, осуществляя некоторую обработку в соответствии с правилами. В ответ на запрос коммутатор может сообщить контроллеру о своих возможностях и конфигурации, а также сигнализировать об изменениях в своем состоянии, например, потерю канала или возникновение ошибки. В остальном, коммутатор полностью полагается на контроллер;
- придорожный сетевой узел (ПСУ): развернутые вдоль дороги устройства, связывающие SDN коммутаторы с конечными пользователями, участниками дорожного движения. ПСУ могут являться маршрутизатором, мостом, точкой доступа и клиентом;
- облако: происходят различные вычисления такие, как расчёт скорости автомобиля, дистанции, оценка дорожной обстановки. Облако вычислений связано с базой данных, которая постоянно обновляется, и использует хранящиеся в базе данных знания для осуществления вычислений;
- база данных: база, в которой хранится информация о дорожной обстановке, автомобилях и их владельцах.

Также для передачи данных на большие расстояния могут быть использованы вышки связи с мощными антеннами 4G/LTE.

Ниже приведена схема, на которой изображена структура сети (рисунок 23):

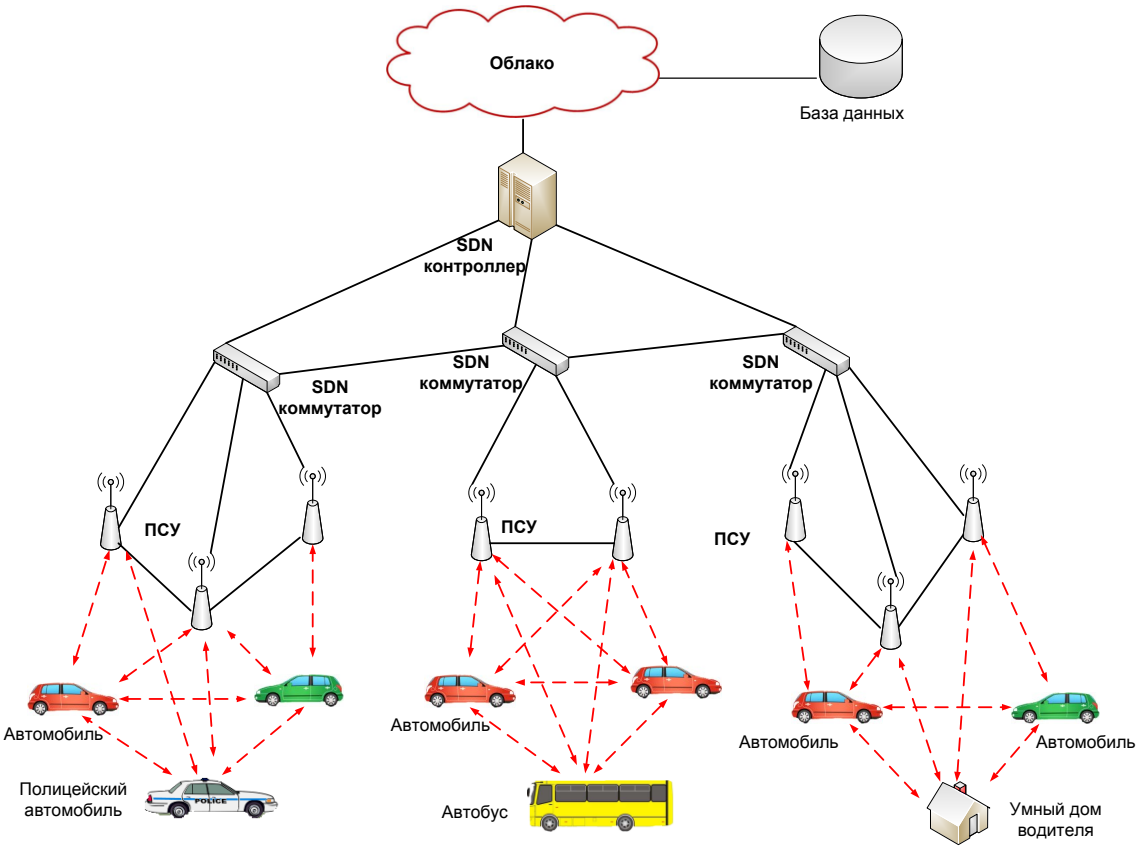

Рисунок 23 – Структура VANET сети, построенной на основе ПКС

На рисунке 24 показаны внутренние компоненты беспроводного узла SDN, который содержит все функциональные возможности коммутатора OpenFlow, а также дополнительный интеллект для управления различными режимами работы в среде VANET. Количество WiFi интерфейсов, используемых в качестве канала передачи данных, зависит от конфигурации и сервиса, который требуется поддерживать SDN коммутатору.

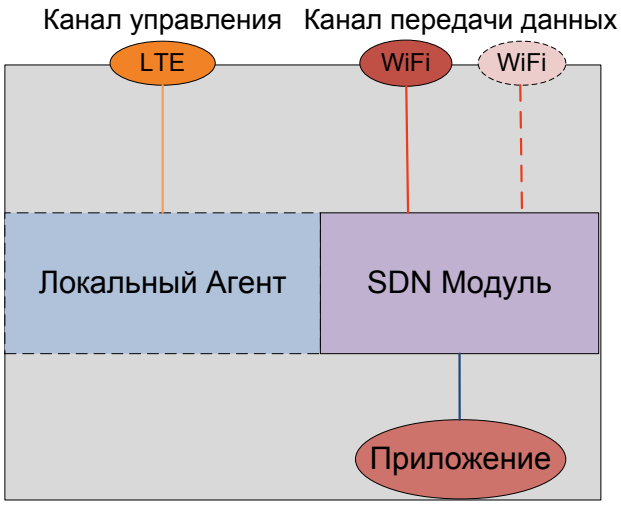

Рисунок 24 – Компоненты коммутатора SDN

В состав каждого беспроводного узла SDN входит локальный агент SDN, функциональность которого зависит от того, какие функции включены на коммутаторе SDN. При потере связи с контроллером SDN локальный агент может служить в качестве контроллера резервного копирования, а также в качестве основного интеллектуального компонента. Традиционные Ad hoc протоколы маршрутизации (например, GPSR, AODV, DSDV и OLSR) поддерживаются в качестве агентов резервных механизмов, чтобы позволить сети SDN вернуться к Ad hoc сети даже в том случае, когда связь с контроллером SDN будет недоступна. В случаях, когда подключение к контроллеру SDN является стабильным и имеет полный контроль, SDN агент выполняет минимальную интеллектуальную нагрузку.

Отличительной характеристикой одноранговых Ad hoc сетей является то, что узлы действуют и как хосты (отправка/прием трафика) и как маршрутизаторы (перенаправление трафика от имени других узлов). Трафик из любого беспроводного узла (например, трафика приложений) будет проходить через свой собственный модуль SDN перед отправкой, что позволяет контроллеру SDN определять доступ пользовательского трафика в сети.

4.4 Режимы работы

Несмотря на то, что концепцией SDN является разделение управления и плоскости данных, существуют различия в том, как программно-конфигурируемая сеть VANET может работать на основе степени управления контроллера SDN. Для рассматриваемых архитектур можно выделить следующие три режима работы:

- Центральный режим: это режим, в котором контроллер SDN контролирует все действия, выполняемые коммутатором SDN и ПСУ. Как показано на рисунке 25, контроллер SDN будет передавать вниз по иерархии все правила того, как следует обрабатывать трафик.

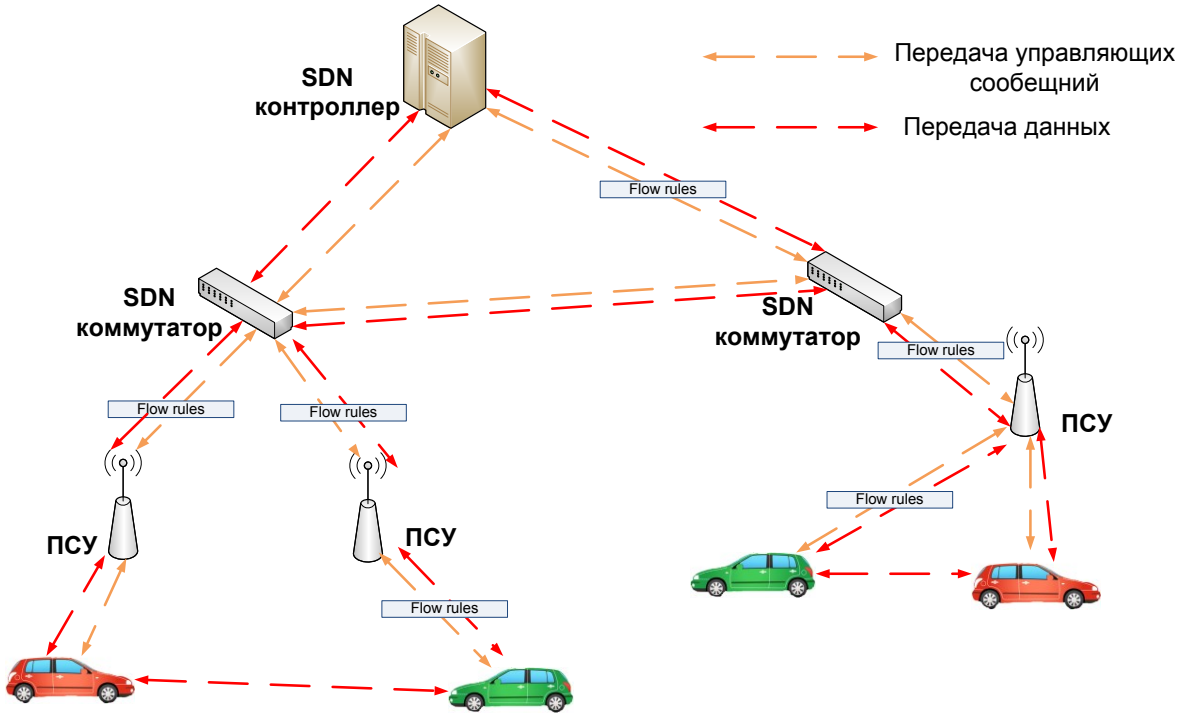

Рисунок 25 – Центральный режим работы сети

- Распределенный режим: это режим, в котором коммутаторы SDN и ПСУ не получают какие-либо указания от контроллера SDN при передаче пакетов данных. Этот режим управления, в сущности, очень похож на оригинальную Ad hoc сеть без каких-либо признаков SDN, за исключением того, что локальный агент на каждом коммутаторе SDN управляет поведением каждого отдельного узла (например, запуска маршрутизации GPSR), как показано на рисунке 26.

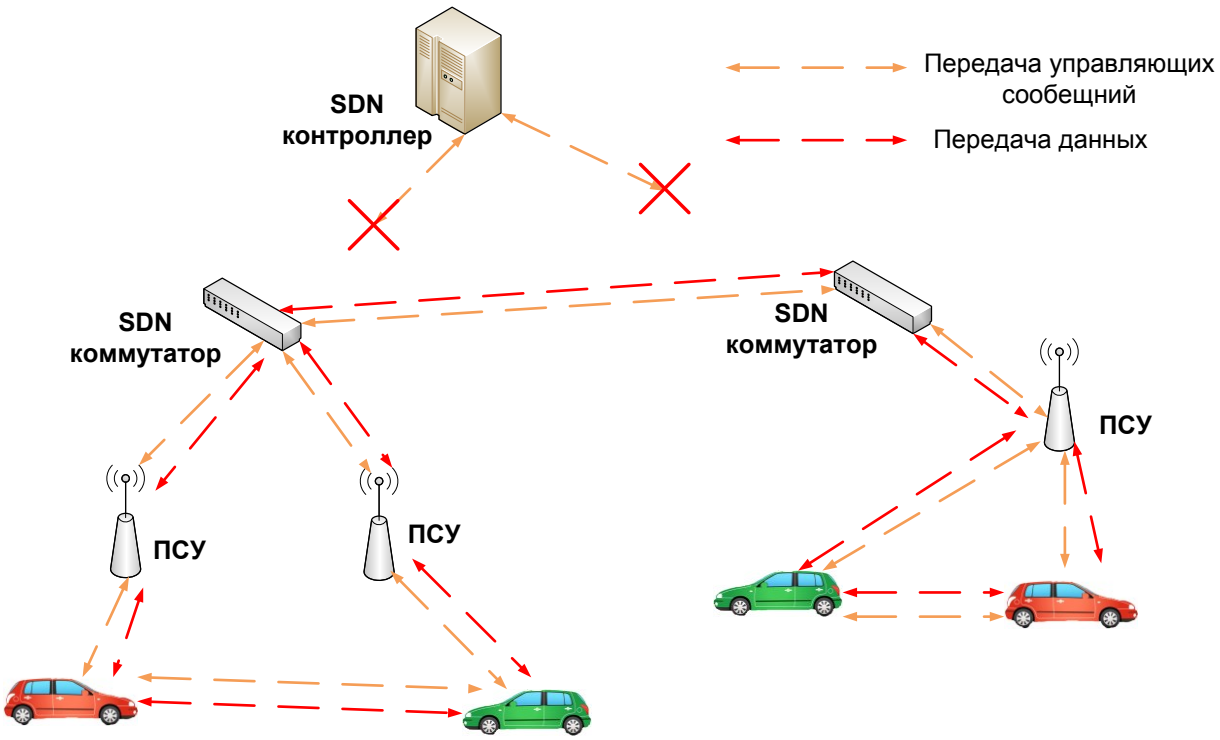

Рисунок 26 – Распределенный режим работы сети

- Гибридный режим: этот режим включает в себя все рабочие режимы системы, в которой контроллер SDN осуществляет контроль в любом месте между полным и нулем. На рисунке 27 показан пример, в котором не имеет место полный контроль со стороны контроллера SDN, то есть часть управления обработкой пакетов передано местным агентам. Поэтому управление обменом трафика распределено между всеми элементами SDN. Например, возможна ситуация, когда вместо того, чтобы посылать полные правила потока, контроллер SDN рассылает правила политики, которые определяют поведение, в то время коммутаторы SDN и ПСУ используют локальный интеллект для пересылки пакетов и обработки уровня потока. Таким образом, контроллер SDN инструктирует коммутаторы SDN и ПСУ для запуска конкретного протокола маршрутизации с определенными параметрами.

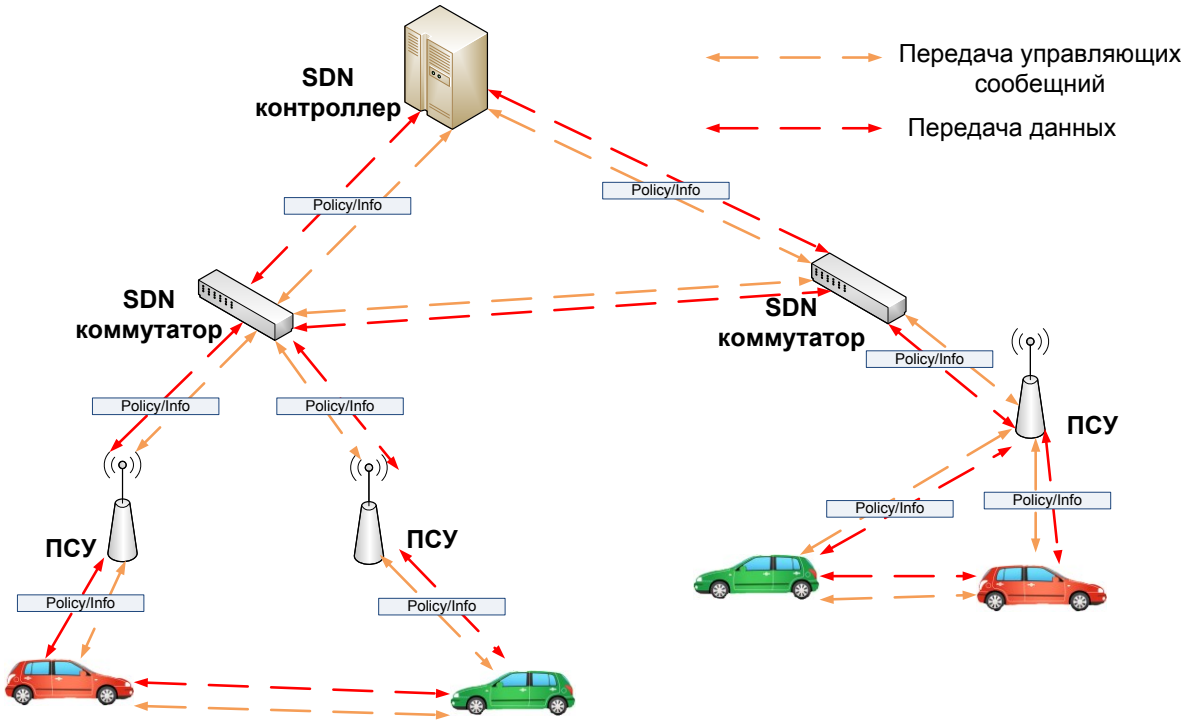

Рисунок 27 – Гибридный режим работы сети

Режим центрального управления ведет себя аналогично проводной архитектуре SDN, где контроллер SDN определяет все правила. Тем не менее, одной из наиболее острых проблем беспроводной связи является ее надежность/доступность, всегда существует вероятность возможной потери связи между мобильными узлами и контроллером. Это является причиной того, почему SDN VANET должны иметь механизмы восстановления после отказа, которые бы гарантировали, что система может функционировать, связь с контроллером SDN потеряна или нарушена. Для этой цели на каждом беспроводном узле SDN находится локальный агент, оснащённый искусственным интеллектом. Например, если связь с контроллером SDN теряется, система может вернуться к управлению традиционный протокол маршрутизации, такой как GPSR.

Изучение топологии сети имеет большое значение для контроллера SDN при принятии разумных решений, поэтому в программно-конфигурируемой сети VANET следует использовать сообщения-маячки, широко распространенную технологию в системах VANET. Каждый беспроводный узел SDN будет обмениваться сообщениями-маячками, чтобы узнать информацию о ближайших

соседях. Эта информация должна постоянно обновляться на контроллере SDN, который в свою очередь использует эти данные для построения таблиц подключенных узлов и принятия решений, таких как выбор пути для маршрутизации пакетов через сеть. Эксплуатируя эту функцию, может обеспечить много преимуществ для управления мобильностью в SDN VANET.

4.5 Преимущества и новые сервисы эластичной ПКС плтаформы

В отделенной плоскости управления SDN обеспечивает гибкость и программируемость сети, что позволяет системе адаптироваться к изменяющимся условиям и требованиям. Эта особенность позволяет программно-конфигурируемой VANET сети принимать более обоснованные решения на основе комбинированной информации из различных источников. Кроме того, динамичность и гибкость позволяют реагировать на внезапные события, например при чрезвычайных ситуациях и изменяющихся требованиях. В этом разделе мы опишем преимущества Software-Defined VANETs, и описать несколько услуг, которые могут быть улучшены за счет использования этих преимуществ.

Классифицируя преимущества от использования программно-конфигурируемых VANET сетей, можно выделить 3 области:

- Выбор пути: Понимание SDN контроллером состояния всей сети позволяет системе принимать более обоснованные решения о маршрутизации. Например, в стандартных VANET сетях трафик данных может стать несбалансированным, либо потому, что самый короткий путь маршрутизации трафика невозможно определить, так как система сфокусирована на некоторых выбранных узлах, или потому, что приложение, работающее с видеоинформацией, требует большую пропускную способность на пути. Когда подобная ситуация обнаруживается контроллером SDN, он может начать процесс перенаправления трафика для улучшения полезность сети и уменьшения количества заторов.

- Выбор частоты/канала: Когда узлы SDN имеют несколько доступных беспроводных интерфейсов или настраиваемых радиостанций, таких как когнитивное радио, программно-конфигурируемая VANET сеть может позволить оптимизировать выбор используемого(-ой) канала/частоты. Например, контроллер SDN может динамически решить в какое время, какой тип трафика будет использовать тот или иной радиоинтерфейс/частоты. Это преимущество может использоваться при резервировании каналов для аварийно-спасательных служб VANET.
- Выбор мощности: контроллер SDN сети VANET будет иметь достаточный объем информации, чтобы определить, следует ли изменить мощности беспроводных интерфейсов, для увеличения/уменьшения дальности передачи данных. Например, контроллер SDN собирает информацию с узлов и определяет, что плотность узлов слишком мала, и выдает соответствующие команды по увеличению мощности, для того чтобы добиться более разумной и стабильной доставки пакетов и уменьшить помехи.

На основе преимуществ, описанных выше, можно выделить ряд услуг, которые могут быть улучшены с помощью программно-конфигурируемых VANET сетей.

- SDN Assisted VANET Safety Service: Повышение безопасности дорожного движения за счет использования V2V связи является одним из основных вариантов использования VANETs. Мы покажем, как Software-Defined VANET может улучшить услуги по сравнению с традиционными методами. SDN могут использоваться для резервирования или ограничения определенных частот, так что только аварийный трафик использует этот зарезервированный путь. Разница с использованием традиционных аварийных каналов заключается в том, что резервирование в SDN архитектуре настраивается динамически. Контроллер SDN может назначить потоки по определенным каналам или удалять их в зависимости от текущих условий дорожного движения и требований приложений. Данная

возможность также может быть использована, чтобы предложить различный уровень услуг на основе политик. Как это может быть сделано путем изменения правил во время периода чрезвычайного положения. Аварийный трафик получает приоритет над остальным трафиком.

- SDN-based On Demand VANET Surveillance Service: служба наблюдения за транспортными средствами является еще одной областью, в которой программно-конфигурируемые VANET сети позволяют добиться ряда преимуществ. В традиционной архитектуре, запрашивающая сторона (например, полицейский автомобиль) должна посылать запрос на данные наблюдения. В системе SDN VANET, этот запрос выполняется контроллером SDN. Контроллер SDN устанавливает правила потока для данных наблюдения, чтобы эти данные достигли всех необходимых узлов. Таким образом, передача данных напоминает широковещательную рассылку, осуществляемую контроллером в отношении группы полицейских автомобилей.
- Wireless Network Virtualization Service: услуги по виртуализации сети призваны обеспечить абстрактные логические сети на общих физических сетевых ресурсах. SDN уже используются в центрах обработки данных для предоставления услуг по виртуализации сети, эту же идею можно применить и для программно-конфигурируемых VANET сетей. Идея заключается в том, чтобы различные потоки данных выбирали различные интерфейсы, используя разные частоты. Если радиочастоты, используемые каждой отдельной сети, неодинаковы, трафик отдельных подсетей является изолированным друг от друга, и таким образом можно эффективно разделить сеть и создать виртуальные беспроводные сети.

4.6 Решение выявленных проблем

Предложенные варианты архитектуры и предлагаемый централизованный подход позволяет решить ряд проблем, описанных в предыдущей главе, а также дает преимущества по сравнению с традиционными транспортными одноранговыми сетями:

- повышение надежности за счет агрегации информации на контроллере. При использовании классической архитектуры на каждом узле хранилась база данных о состоянии каналов, маршрутов и узлов, однако использование централизованного подхода позволяет собирать информацию в одном месте – на контроллере. Таким образом, такая централизованная база данных будет содержать значительно меньше несогласованной информации, и такой подход позволит уменьшить вероятность возникновения циклов в сети;
- упрощение структуры и логики сетевых устройств, так как при централизованном подходе не требуется обрабатывать большое количество стандартов и протоколов, а достаточно выполнять только инструкции, полученные от контроллера;
- повышение безопасности передаваемых данных за счет возможности создания политики безопасности на контроллере, обеспечения надежной идентификации, размещения межсетевых экранов и сетевых антивирусов;
- программируемость и гибкость управления сетью, а также значительное упрощение возможности модификации управления сетью за счет создания новых приложений или модификации существующих. Кроме того централизованный подход позволяет повысить уровень автоматизации управления и удобство администрирования сетей;
- адаптивность управления сетью, то есть возможность изменять поведение и состояние сети в режиме реального времени с учетом изменяющихся условий функционирования. Также появляется возможность адаптироваться к меняющимся потребностям пользователей сетей за счет создания новых сетевых приложений и сервисов. Кроме того на разработку

сетевых приложений требуется значительно меньше времени по сравнению с переконфигурированием всей сети в ручном режиме;

- независимость от оборудования и проприетарного программного обеспечения производителей сетевого оборудования;
- возможность независимого развертывания и масштабирования уровня управления и уровня передачи данных, что повышается удобство обслуживания транспортной сети;
- снижение стоимости коммутаторов и сетевой инфраструктуры в целом за счет вынесения интеллекта на контроллер.

Таким образом, подход SDN позволяет значительно автоматизировать и упростить управление сетями за счет возможности их «программирования», позволяя строить гибкие масштабируемые сети, которые могут легко адаптироваться к изменяющимся условиям функционирования и потребностям пользователей. Внедрение этого подхода, должно оказать значительное влияние на управление сетевой инфраструктурой в транспортных сетях VANET.

5 МЕТОД АНАЛИЗА РИСКОВ КИБЕРБЕЗОПАСНОСТИ АКТИВОВ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

Анализ риска обеспечивает базу для оценивания риска, мероприятий по снижению риска и принятия риска для объектов КИИ. Анализ риска включает в себя выявления угроз, методы анализа и методики управления рисками. Идентификация и анализ рисков являются ключевым элементом процесса управления риском. Понятие анализа риска применяется практически во всех областях. В рамках исследования будем придерживаться определений, представленных в стандартах ГОСТ Р ИСО 31000 «Менеджмент риска. Принципы и руководство» и ГОСТ Р ИСО/МЭК 31010 «Менеджмент риска. Методы оценки риска».

Риск – следствие влияния неопределенности на достижение поставленных целей. Под следствием влияния неопределенности необходимо понимать положительное или отрицательное отклонение от ожидаемого результата или события. Цели могут быть различными по содержанию и назначению.

Менеджмент риска – скоординированные действия по руководству и управлению организацией в области риска.

Оценка риска – процесс, охватывающий идентификацию риска, анализ риска и сравнительную оценку риска.

Анализ риска – процесс изучения природы и характера риска и определения уровня риска. Анализ риска обеспечивает базу для проведения сравнительной оценки риска и принятия решения об обработке риска.

Уровень риска – мера риска или комбинации нескольких видов риска, характеризуемая последствиями и их правдоподобностью/вероятностью.

Допустимый риск – риск, который организация и причастные стороны готовы сохранять после обработки риска для достижения своих целей.

Согласно ГОСТ Р ИСО/МЭК 31010, известно множество методов анализа рисков (аналитические, причинно-следственные, формальные).

Мозговой штурм представляет собой обсуждение проблемы группой специалистов в доброжелательной манере, целью которого является идентификация возможных видов отказов и соответствующих опасностей, риска, критериев принятия решений и/или способов обработки риска. Метод предполагает стимулирование обсуждения, периодическое направление обсуждения группы в смежные области и обеспечение охвата проблем, выявленных в результате обсуждения. Данный метод часто применяется в производстве связанно с новыми технологиями, поскольку осуждаемая область еще не полностью изучена. Также данный метод можно легко и быстро применить в реальных условиях работы. Данный метод подходит для рассмотрения модели цифровой платформы CML-Bench, так как такие системы является сравнительно новой технологией и постоянно исследуемой в современном мире информационной безопасности. Поэтому возникающие новые инциденты безопасности (0-day) могут быть отражены оперативно собравшейся группой экспертов.

Структурированные или частично структурированные интервью является методом, в котором опрашиваемому задают вопросы из заранее подготовленного перечня, поощряющие всесторонний анализ ситуации и, таким образом, более полную идентификацию опасностей и риска. Частично структурированное интервью аналогично структурированному, однако оно обеспечивает большую свободу при обсуждении исследуемой проблемы. Данный метод не подходит для рассматриваемой модели, поскольку он требует больших затрат времени, что в критической ситуации может быть решающим фактором.

Метод Дельфи предназначен для получения обобщенного мнения группы экспертов. Существенной особенностью метода Дельфи является то, что эксперты выражают свое мнение индивидуально и анонимно, при этом имея возможность узнать мнения других экспертов. Данный метод не подходит для

рассматриваемой модели, поскольку в анонимности экспертов нет необходимости.

Контрольные листы представляют собой перечни опасностей, риска или отказов средств управления, которые обычно разрабатывают на основе полученного ранее опыта, результатов предыдущей оценки риска или результатов отказов, произошедших в прошлом. Данный метод не подходит для рассматриваемой модели, т.к. отсутствует опыт и результаты предыдущей оценки рисков.

Предварительный анализ опасностей является простым индуктивным методом анализа, цель которого состоит в идентификации опасностей, опасных ситуаций и событий, которые могут нарушить работу или нанести вред данному виду деятельности, оборудованию или системе. Данный метод можно применить на начальном этапе разработке, с целью заложить фундамент безопасности системы.

FMEA (анализ видов и последствий отказов) – метод, используемый для идентификации возможных путей отказа компонентов, систем и/или процессов, которые могут привести к невыполнению их функций. Указанный метод позволяет идентифицировать все разновидности отказов различных компонентов системы, рассмотреть последствия и механизмы отказа, найти пути достижения безотказной работы и/или смягчения последствий отказа. Метод подходит для рассматриваемой системы.

FMECA (анализ критичности видов и последствий отказов) – расширенная и дополненная версия метода FMEA, позволяющая оценить критичность и значимость каждого найденного вида отказа. Данный метод подходит для рассматриваемой модели системы.

SA (анализ скрытых дефектов) – метод идентификации ошибок проектирования. Скрытыми дефектами могут являться неявные дефекты вычислительной техники и/или программного обеспечения, которые могут

вызвать нежелательное поведение или препятствовать наступлению ожидаемого события и при этом не являются следствием отказа компонентов. Эти дефекты имеют случайный характер и могут быть не обнаружены во время испытаний и тестирования. Скрытые дефекты могут приводить к некорректному выполнению технологических процессов и отказу системы. Входными данными указанного метода являются результаты проведения комплексных проверок на основе других методов. В связи с трудоемкостью указанного метода, предпочтение отдается другим, более простым методам.

Метод FTA основан на идентификации и анализе факторов, способных провоцировать возникновение нежелательного события. Путем дедукции производится идентификация, выстраивание логическим образом и представление на диаграмме в виде дерева исследуемых факторов. Диаграмма отображает эти факторы и их логическую связь с конечным событием. Данный метод не подходит для рассматриваемой системы CML-Bench, поскольку возможные инциденты безопасности могут быть не изучены, из-за чего построить дерево событий не представляется возможным.

Метод ETA основан на графическом представлении взаимоисключающих последовательностей событий, последующих появлению исходного события.

Метод ETA может быть применен для качественной и/или количественной оценки. Данный метод не подходит для рассматриваемой системы, поскольку он применим только для двух состояний системы (работоспособного состояния и отказа).

Метод HAZOP (исследование опасности и работоспособности) подобен методу FMEA и направлен на идентификацию путей отказа процесса, системы или процедуры, их причин и последствий. Отличие от метода FMEA заключается в рассмотрении в методе HAZOP нежелательных результатов и отклонений от заданных целей, условий возможных причин и видов отказа. Данный метод не

подходит для рассматриваемой модели, поскольку он требует наличие подробной документации и детально описанных требований к системе.

Метод LOPA направлен на анализ достаточности мер по управлению или снижению риска. Обычно, данный метод применяют для достижения большей точности после применения метода HAZOP или PHA. В силу отсутствия подробной документации и детально описанных требований метод LOPA неприменим к рассматриваемой системы.

Метод Bow-Tie («галстук-бабочка») основан на схематическом описании и анализе пути развития опасного события от причин до последствий. Данный метод включает в себя исследование причин события при помощи дерева отказов и анализ последствий при помощи дерева событий. Основное внимание при использовании данного метода сфокусировано на барьерах между причинами, опасными событиями и последствиями. Данный метод не подходит для рассматриваемой системы, поскольку он не позволяет отображать совокупности причин, возникающих одновременно и вызывающих последствия.

В таблице 5 представлен анализ рисков кибербезопасности КИИ методом FMECA.

Анализ критичности в таблице является качественным, но может быть количественным при использовании показателя фактического процента отказов. Таблица включает в себя ранга приоритетности риска (RPN, Risk Priority Number). Его рассчитывают путем умножения ранга значимости последствий отказа (от 1 до 10) на вероятность отказа и возможность выявления проблемы.

Выходными данными метода FMECA являются перечень видов отказа, механизмов возникновения отказа, его последствий для каждого компонента системы, ранжирование значимости отказов и рекомендации по предотвращению и устранению отказов.

Таблица 5 – Анализ рисков кибербезопасности КИИ методом FMECA

Название	Функция	Отказ	Причина отказа	Последствия	Серьезность	Частота	Обнаружение	RPN	Комментарий
Сервер базы данных (PostgreSQL)	Хранение конфиденциальных данных	Утечка конфиденциальных данных	Уязвимости ПО, некорректная конфигурация ПО, несанкционированный физический доступ	Утечка конфиденциальной информации	6	4	5	120	Мониторинг трафика, контроль за трафиком, корректная настройка и своевременное обновление ПО, контроль доступа к оборудованию
Сервер управления очередью (RabbitMQ)	Передача асинхронных сообщений между сервисами Платформы	Отсутствие возможности передачи сообщений	MITM атаки, уязвимости ПО, атака на радиоканал обмена сообщениями	Нарушение функционирования платформы	5	5	7	175	Наличие резервных каналов связи
Центральный сервер	Управление всеми базовыми сущностями системы, пользователями и разграничение доступа (сервис Core)	Прекращение передачи данных	Уязвимости ПО, устаревшая версия ПО, проблемы с электропитанием, неверная сетевая конфигурация, физический доступ злоумышленника	Прекращение функционирования платформы	10	6	7	420	Защитные механизмы (программные и аппаратные) по предотвращению атак, обеспечение бесперебойного электропитания, методы ограничения физического доступа к машине посторонних лиц.
	Управление запросами на изменения и сообщениями об ошибках от пользователей (сервис Helpdesk)	Прекращение сбора данных со всех сервисов	Проведение атак на сетевом уровне	Нарушение работоспособности сети	10	4	3	120	Защитные механизмы (программные и аппаратные) по предотвращению атак
	Координация	Нарушение	Проведение атак на	Изменение	8	4	10	320	Защитные механизмы

	внутреннего взаимодействия между сервисами (сервис Discovery)	межсервисного взаимодействия	сетевом уровне	топологии, увеличение времени передачи					(программные и аппаратные) по предотвращению атак
Сервер сбора статистика	Управление очередью задач (сервис Queue)	Прекращение передачи	Устаревшая версия ПО, проблемы с неверная сетевая конфигурация	Нарушение функционирования платформы	6	2	4	48	Корректная настройка и своевременное обновление ПО
	Сбор параметров запускаемых задач (сервис Task statistics)	Прекращение сбора данных	Уязвимости ПО, устаревшая версия ПО, неверная сетевая конфигурация	Нарушение функционирования платформы	3	1	5	15	Защитные механизмы (программные и аппаратные) по предотвращению атак
	Управления лицензиями на запускаемые расчетные пакеты (сервис Licenses manager)	Невозможность подключения новых серверов лицензий	Уязвимости ПО, устаревшая версия ПО, неверная сетевая конфигурация	Нарушение функционирования платформы	4	2	5	40	Корректная настройка и своевременное обновление ПО
Сервер управления кластерами	Запуск задач и управление вычислительными ресурсами (сервис Clusters manager)	Невозможность запуска задач	Уязвимости ПО, устаревшая версия ПО	Нарушение функционирования платформы	5	2	5	50	Корректная настройка и своевременное обновление ПО
Сетевое хранилище	Размещение расчетных моделей, а также результатов расчетов	Утечка конфиденциальных данных	Уязвимости ПО, устаревшая версия ПО, неверная сетевая конфигурация	Утечка конфиденциальной информации	7	6	5	210	Защитные механизмы (программные и аппаратные) по предотвращению атак, корректная настройка и своевременное обновление ПО
Вычислитель	Проведение вычисле	Невозможность	Проблемы с электропитанием,	Нарушение функцио	2	2	3	12	Корректная настройка и

ные узлы	ний и расчетов моделей	провед ения вычисл ений	неверная сетевая конфигурац ия	нирован ия платфор мы					своевременное обновление ПО

Согласно ГОСТ Р ИСО/МЭК 31010, существуют различные формальные (статистические) методы.

Марковские диаграммы применимы в ситуации, когда будущее состояние системы зависит только от ее текущего состояния. Данный метод обычно используют для анализа ремонтопригодных систем, которые могут работать во многих режимах, и в ситуациях, когда применение анализа надежности отдельных блоков системы нецелесообразно. Данный метод не подходит для КИИ, поскольку он основан на предположении о постоянстве вероятностей перехода и наличием только двух возможных состояний элементов системы (отказа и восстановления).

Метод Монте-Карло является способом оценки влияния неопределенности оценки параметров системы в широком диапазоне ситуаций. Метод обычно используют для оценки диапазона изменения результатов и относительной частоты значений в этом диапазоне для количественных величин, таких как стоимость, продолжительность, производительность, спрос и др. Данный метод не подходит для рассматриваемой системы, поскольку он предполагает, что неопределенность данных можно описать известным распределением, а распределение неопределенность данных для данной модели неизвестно.

Байесовский анализ представляет собой графическую модель, представляющую переменные и их вероятностные взаимосвязи. Сеть состоит из узлов, представляющих случайные переменные, и стрелок, связывающих родительский узел с дочерним узлом (родительский узел - переменная, которая непосредственно влияет на другую дочернюю переменную). Данный метод подходит для рассматриваемой модели, поскольку графические представления выходных данных обеспечивают простоту понимания модели, при этом данные

могут быть легко изменены для исследования корреляции и чувствительности параметров. Он содержит несколько типов связей:

1. Последовательная связь
2. Сходящаяся связь
3. Расходящаяся связь

Рассмотрим каждую связь подробнее.

Пусть в связи участвуют 3 переменных: x, y и z. Переменная x влияет на y, а y, в свою очередь, влияет на z. В таком случае вероятностное разложение будет выглядеть следующим образом: $p(x, y, z) = p(x) \cdot p(y|x) \cdot p(z|y)$. Граф, в таком случае, будет выглядеть следующим образом (рисунок 28):

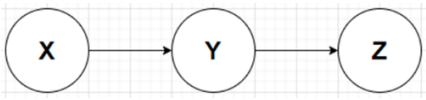

Рисунок 28 - Граф последовательной связи байесовской сети анализа рисков

В данном типе связи переменные x и y влияют на z. Тогда вероятностное разложение будет иметь вид $p(x, y, z) = p(x) \cdot p(y) \cdot p(z|x, y)$, а граф будет следующим (рисунок 29):

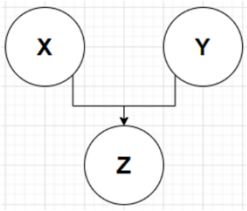

Рисунок 29 – Граф сходящейся связи байесовской сети анализа рисков

При данном типе связи в зависимости от состояния x выполняется переход к y или z. Получим следующее разложение $p(x, y, z) = p(x) \cdot p(y|x) \cdot p(z|x)$ и граф (рисунок 30):

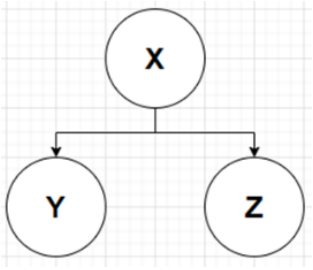

Рисунок 30 – Граф расходящейся связи байесовской сети

На основе данных формул посчитаем вероятности перехода между блоками и вероятности возникновения определенных событий.

$$P_5 = P_{1-5} = 0{,}2$$

$$P_6 = P_{1-6} = 0{,}15$$

$$P_7 = P_{1-7} * P_{3-7} = 0{,}15 * 0{,}04 = 0{,}006$$

$$P_8 = P_{4-8} = 0{,}5$$

$$P_{10} = P_5 * P_{5-10} * P_8 * P_{8-10} = 0{,}2 * 0{,}45 * 0{,}5 * 0{,}39 = 0{,}01755$$

$$P_9 = P_{2-9} = 0{,}25$$

$$\begin{aligned} P_{11} = P_7 * P_{7-11} * P_9 * P_{9-11} * P_8 * P_{8-11} * P_6 * P_{6-11} * P_{2-11} \\ = 0{,}006 * 0{,}004 * 0{,}25 * 0{,}15 * 0{,}5 * 0{,}09 * 0{,}15 * 0{,}18 * 0{,}15 \\ = 4{,}1009 * 10^{-9} \end{aligned}$$

$$\begin{aligned} P_{12} = P_{10} * P_{10-12} * P_9 * P_{9-12} * P_9 * P_{9-12} * P_{11} * P_{11-12} \\ = 0{,}01755 * 0{,}35 * 0{,}25 * 0{,}17 * 4{,}1009 * 10^{-9} * 0{,}3 \\ = 0{,}3212 * 10^{-12} \end{aligned}$$

$$\begin{aligned} P_{13} = P_{11} * P_{11-13} * P_9 * P_{9-13} = 4{,}1009 * 10^{-9} * 0{,}6 * 0{,}25 * 0{,}22 \\ = 0{,}1353 * 10^{-9} \end{aligned}$$

$$\begin{aligned} P_{14} = P_{11} * P_{11-14} * P_9 * P_{9-14} = 4{,}1009 * 10^{-9} * 0{,}4 * 0{,}25 * 0{,}03 \\ = 0{,}123 * 10^{-10} \end{aligned}$$

На рисунке 31 представлен анализ рисков байесовским методом для КИИ. Входными узлами, помеченные голубым цветом, графа являются потенциальные угрозы цифровой платформы, бирюзовым цветом помечены промежуточные

этапы, красным цветом помечены последствия реализации потенциальных угроз. Вероятности перехода в графе определяются экспертной оценкой.

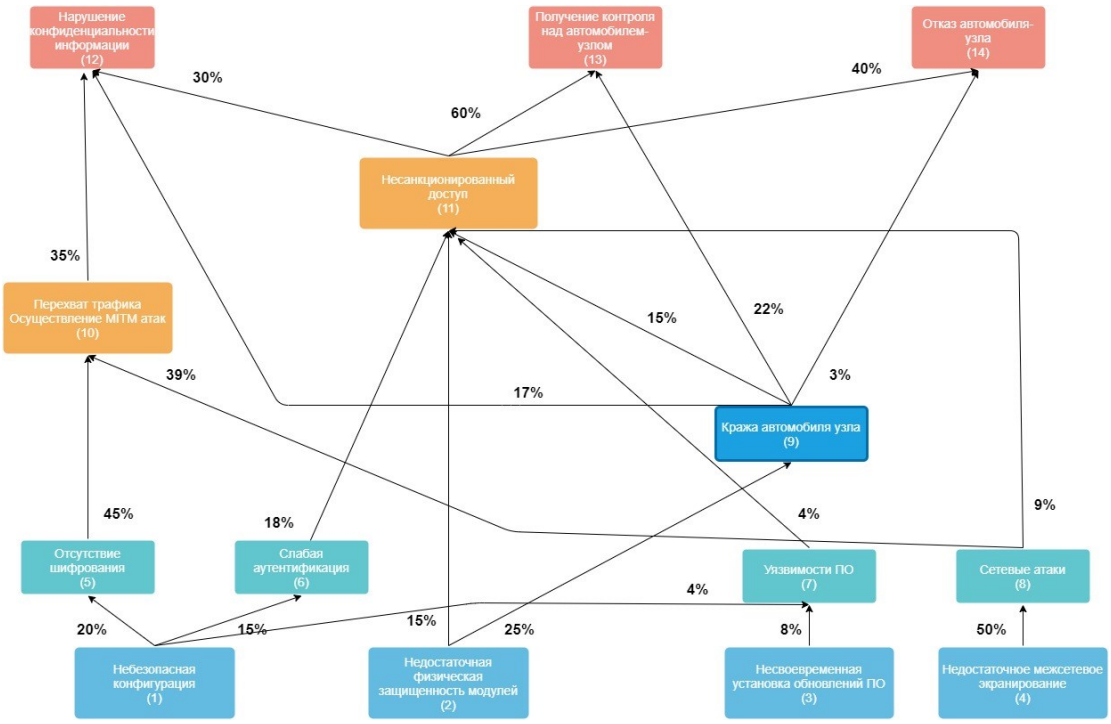

Рисунок 31 – Анализ рисков байесовским методом

После определения вероятностей риска можно сделать вывод, что почти все вероятности имеют одинаковое значение с незначительным различием.

По результатам оценки рисков можно сделать вывод, что в первую очередь необходимо устранять уязвимости ПО, некорректную конфигурацию ПО, несанкционированный физический доступ. Для устранения таких угроз необходимо контролировать сетевой трафик, осуществлять корректную настройку и своевременное обновление ПО, а также контролировать доступ к оборудованию.

Исходя из проведенного анализа, смоделировано влияние на уровень риска принятие мер по снижению выявленных рисков:

– автоматическая установка обновления ПО;

– использование TLS;

– установка сигнализации на узлы для понижения вероятности кражи устройства;

– настройка аутентификации и авторизации на узлах, в том числе пароли устройств должны быть сложными (более 8 символом, содержащие строчные и заглавные буквы, а также символы);

– не использовать стандартные и легко подбираемые пароли из словарей;

– дополнительная физическая блокировка узлов, например, использование кроме сигнализации и ключа с паролем обычного ключа;

– обязательное использование межсетевого экрана.

Ниже представлена оценка рисков байесовским методом после внесенных изменений.

$$P_5 = P_{1-5} = 0{,}2$$

$$P_6 = P_{1-6} = 0{,}15$$

$$P_7 = P_{1-7} * P_{3-7} = 0{,}003 * 0{,}04 = 0{,}000012$$

$$P_8 = P_{4-8} = 0{,}15$$

$$P_{10} = P_5 * P_{5-10} * P_8 * P_{8-10} = 0{,}2 * 0{,}25 * 0{,}15 * 0{,}21 = 0{,}001575$$

$$P_9 = P_{2-9} = 0{,}05$$

$$\begin{aligned} P_{11} = {} & P_7 * P_{7-11} * P_9 * P_{9-11} * P_8 * P_{8-11} * P_6 * P_{6-11} * P_{2-11} \\ & = 0{,}000012 * 0{,}004 * 0{,}05 * 0{,}15 * 0{,}15 * 0{,}03 * 0{,}15 * 0{,}18 * 0{,}03 \\ & = 1{,}13122 * 10^{-15} \end{aligned}$$

$$\begin{aligned} P_{12} = {} & P_{10} * P_{10-12} * P_9 * P_{9-12} * P_{11} * P_{11-12} \\ & = 0{,}001575 * 0{,}35 * 0{,}05 * 0{,}17 * 1{,}13122 * 10^{-15} * 0{,}15 \\ & = 0{,}92227 * 10^{-23} \end{aligned}$$

$$\begin{aligned} P_{13} = {} & P_{11} * P_{11-13} * P_9 * P_{9-13} = 1{,}13122 * 10^{-15} * 0{,}3 * 0{,}05 * 0{,}22 \\ & = 0{,}3733 * 10^{-11} \end{aligned}$$

$$\begin{aligned} P_{14} = {} & P_{11} * P_{11-14} * P_9 * P_{9-14} = 1{,}13122 * 10^{-15} * 0{,}2 * 0{,}05 * 0{,}03 \\ & = 0{,}339 * 10^{-12} \end{aligned}$$

Можно сделать вывод, что принятые меры дают определенный результат, и расчетные риски снижаются.

ЗАКЛЮЧЕНИЕ

В результате проведённого диссертационного исследования была достигнута цель, а именно - разработка подхода, обеспечивающего безопасность объектов КИИ.

В рамках диссертационного исследования автором решены следующие задачи:

1. Выявлены особенности объектов критической информационной инфраструктуры и применяемых в них механизмов обеспечения безопасности.
2. Построена модель типового объекта критической информационной инфраструктуры.
3. Предложен подход к выбору защитных мер, основанный на нейронечеткой модели.
4. Разработан метод выбора ресурсов платофрмы эластичных вычислений.
5. Разработан метод оценки защищенности активов критической информационной инфраструктуры и выработки контрмер по снижению киберрисков.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Piggin R. S. H. (2013). Process safety and cyber security convergence: Lessons identified, but not learnt? In: IET Conference Proceedings. The Institution of Engineering & Technology
2. 2. Zimmermann A, Schmidt R, Jugel D, Möhring M. Evolving enterprise architectures for digital transformations. DEC. 2015;15:25-26
3. 3. Bryant IRC. Towards a trustworthy software ecosystem. International Software Quality Management Conference (SQM); 2012. pp. 1-7

4. 4. Kriaa S, Pietre-Cambacedes L, Bouissou M, Halgand Y. A survey of approaches combining safety and security for industrial control systems. Reliability Engineering & System Safety. 2015;139:156-178

5. 5. Carlsson B, Jacobsson A. (2012). Om säkerhet i digitala ekosystem. Studentlitteratur AB, Lund, Sverige

6. 6. Longstaff PH, Armstrong NJ, Perrin K, Parker WM, Hidek MA. Building resilient communities: A preliminary framework for assessment. Homeland Security Affairs. 2010;6(3):1-22

7. 7. Sheard S, Mostashari A. A framework for system resilience discussions. In: Proc Eighteenth Annual Int Symp INCOSE; 2008

8. 8. Merkow MS, Raghavan L. Secure and Resilient Software Development. CRC Press; London, UK, 2010

9. 9. Sharkov G. From cybersecurity to collaborative resiliency. In: Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense, ACM; 2016, October. pp. 3-9

10. 10. Koscher K, Czeskis A, Roesner F, Patel S, Kohno T, Checkoway S, Savage S. Experimental security analysis of a modern automobile. In: 2010 IEEE Symposium on Security and Privacy; 2010, May. pp. 447-462

11. 11. Lima A, Rocha F, Völp M, Esteves-Veríssimo P. Towards safe and secure autonomous and cooperative vehicle ecosystems. In: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy, ACM; 2016. pp. 59-70

12. 12. DHS. Department of Homeland Security, Office of Cyber and Infrastructure Analysis: The Future of Smart Cities: Cyber-Physical Infrastructure Risk. 2015

13. 13. Cerrudo C. (2015). An Emerging US (and World) Threat: Cities Wide Open to Cyber Attacks. Securing Smart Cities – White paper – IOActive. www.ioactive.com/pdfs/IOActive_HackingCitiesPaper_cyber-security_CesarCerrudo.pdf

14. 14. Axelsson J, Papatheocharous E, Andersson J. Characteristics of software ecosystems for federated embedded systems: A case study. Information and Software Technology. 2014;56(11):1457-1475

15. 15. Зорина Т.Ю. Методы оценки рисков ИТ-проектов. Учебник / Зорина Т.Ю. –Москва: «Высшее образование в России», 2014. –12с. 22. IT риски [Электронный ресурс]. – Режим доступа: https://www.risk-academy.ru/portfolio-items/it-risks/. (Дата обращения: 07.06.2019 г.).

16. 16. Медведовский И. Современные методы и средства анализа и контроля рисков информационных систем компаний [Электронный ресурс]. – Режим доступа: http://citforum.ru/products/dsec/itrisk/. (Дата обращения 07.06.2019 г.).

17. 17. Методики анализа и оценки рисков информационной безопасности [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/metodiki-analiza-i-otsenki-riskov-informatsionnoy-bezopa

18. snosti. (Дата обращения: 07.06.2019 г.).

19. 18. Методологии управления ИТ-рисками [Электронный ресурс]. – Режим доступа: https://www.osp.ru/os/2006/08/3584582. (Дата обращения: 07.06.2019 г.).

20. 19. Оценка риска технических систем. Основы методологии анализа и управления риском [Электронный ресурс]. – Режим доступа: http://fb.ru/article/332477/otsenka-riska-tehnicheskih-sistem-osnovyi-metodologii-analiza-

21. i-upravleniya-riskom. (Дата обращения: 07.06.2019 г.).

22. 20. Оценка риска: понятие и место в обеспечении безопасности технических систем [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/5996807/page:13. (Дата обращения: 07.06.2019 г.)

23. 21. Peltier T. Facilitated Risk Analysis Process (FRAP) [Электронный ресурс]. – Режим доступа: http://www.peltierassociates.com/frap.htm. (Дата обращения: 13.05.2019 г.).

24. 22. Caralli R., Stevens J., Young L., Wilson W. Introducing OCTAVE Allegro: Improving the Information Security Risk Assesment Process [Электронный ресурс]. – Режим доступа: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=8419. (Дата обращения: 07.06.2019 г.).

25. 23. RiskWatch International: Risk Assesment Software [Электронный ресурс]. – Режим доступа: https://riskwatch.com/. (Дата обращения: 10.05.2019 г.).

26. 24. Астахов А.М. Искусство управления информационными рисками. –М.: Изд-во: ДМК Пресс, 2010. – 312 с

27. 25. Воронцовский А.В. Оценка рисков: учебник и практикум для бакалавриата и магистратуры / А. В. Воронцовский. – М.: Изд-во: Юрайт, 2018. –179 с.

28.