ВВЕДЕНИЕ  
Раздел 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АНАЛИЗА АВИАЦИОННЫХ  
ПРОИСШЕСТВИЙ  
1.1 Понятие и виды авиационных происшествий  
1.2 Анализ статистики авиационных происшествий  
1.3 Организационно-правовое обеспечение процесса анализ авиационных  
происшествий  
1.4 Системы поддержки принятия решений при анализе авиационных  
происшествий  
1.5 Постановка задачи  
Раздел 2. МОДЕЛИ И АЛГОРИТМЫ УСТАНОВЛЕНИЯ ПРИЧИННО-  
СЛЕДСТВЕННЫХ СВЯЗЕЙ ПРИ АНАЛИЗЕ АВИАЦИОННЫХ  
ПРОИСШЕСТВИЙ  
2.1 Гипотеза моделирования  
2.2 Формализация и классификация переменных  
2.3 Математическая модель анализа авиационных происшествий  
2.4 Алгоритм установления причинно-следственных связей  
Раздел 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
3.1 Функциональные требования к программе и обоснование выбора среды  
программирования  
3.2 Создание программного продукта  
3.3 Анализ результатов  
ВЫВОДЫ  
ЛИТЕРАТУРА  
ПРИЛОЖЕНИЯ

**УКАЗАТЕЛЬ ОПРЕДЕЛЕНИЙ, ОБОЗНАЧЕНИЙ И**

**СОКРАЩЕНИЙ**

АП – Авиационное происшествие

ППС – Причинно-следственная связь

БСД – Байесовские сети доверия

ВС – воздушных судов

УВД – управления воздушным движением

СУБП – системы управления безопасностью полетов

BAAA – ( от англ. Bureau of Aircraft Accidents Archives) Бюро авиационных происшествий

ИКАО – (от англ. ) международной гражданской авиации

МАК – Межгосударственный авиационный комитет

СППР – Системы поддержки принятия решений

…

**ВВЕДЕНИЕ**

Гражданская авиация является одной из наиболее динамично развивающихся и технологически сложных отраслей современной экономики, играя ключевую роль в глобальной транспортной системе, деловых связях и туризме. Однако неотъемлемой частью авиационной деятельности являются риски, связанные с безопасностью полетов.

Безопасность полетов – приоритетная задача современной авиации. Несмотря на постоянное совершенствование авиационной техники, внедрение передовых технологий и строгие регламенты, авиационные происшествия по-прежнему случаются, приводя к трагическим последствиям и значительным экономическим потерям. Эффективный анализ АП, выявление их коренных причин и разработка мер по предотвращению повторения подобных инцидентов являются ключевыми элементами обеспечения высокого уровня безопасности полетов.

Особую обеспокоенность вызывает тот факт, что значительная доля авиационных происшествий (по некоторым оценкам, до 80%) связана с так называемым «человеческим фактором» – ошибками, допускаемыми персоналом на различных этапах: от проектирования и технического обслуживания воздушных судов до планирования, управления воздушным движением и непосредственного пилотирования. При этом на действия персонала существенное влияние оказывают внешние условия (метеорология, состояние аэродромной инфраструктуры), техническое состояние ВС, а также организационные аспекты деятельности авиапредприятий и надзорных органов (качество процедур, подготовка кадров, культура безопасности). Выявление сложных, зачастую скрытых, взаимосвязей между этими разнородными факторами и их влияния на исход авиационного события (инцидент, авария, катастрофа) представляет собой актуальную и сложную научно-практическую задачу.

Традиционные методы анализа АП, основанные на статистике и описательном анализе отдельных случаев, не всегда позволяют количественно оценить влияние комбинаций факторов и выявить неочевидные зависимости. В последние годы для решения подобных задач все активнее применяются методы вероятностного моделирования, в частности, Байесовские сети доверия. БСД предоставляют мощный математический аппарат для представления знаний об исследуемой области в виде графа причинно-следственных связей и таблиц условных вероятностей, позволяя не только анализировать исторические данные, но и выполнять логический вывод, оценивая вероятность исхода при различных сценариях и условиях.

Таким образом, *актуальность* данного дипломного исследования обусловлена:

* Необходимостью повышения уровня безопасности полетов в гражданской авиации.
* Высокой долей влияния человеческого, организационного, технического и средового факторов на возникновение АП.
* Потребностью в разработке и применении современных математических методов для анализа сложных причинно-следственных связей и оценки рисков в авиации.
* Перспективностью использования аппарата Байесовских сетей доверия для моделирования и анализа авиационных происшествий.

**Цель дипломной работы:** Разработка и апробация математической модели на основе Байесовской сети доверия для анализа и количественной оценки влияния различных факторов на возникновение авиационных происшествий с самолетами.

Для достижения поставленной цели были решены следующие задачи:

1. Проведен анализ существующих методов расследования и анализа авиационных происшествий, выявлены их преимущества и недостатки.
2. Определен и формализован перечень ключевых факторов (метеорологических, технических, организационных, человеческих, контекстуальных), влияющих на безопасность полетов, и их возможные состояния.
3. Сформирован набор исходных данных на основе анализа отчетов и донесений о реальных авиационных происшествиях и инцидентах с самолетами ГА РФ за период 2012-2023 гг.
4. Разработана математическая модель на основе Байесовской сети доверия, отражающая предполагаемые причинно-следственные связи между идентифицированными факторами и тяжестью исхода АП.
5. Создан программный модуль на языке Python с использованием библиотеки pgmpy для построения структуры БСД, подготовки данных, валидации и обучения параметров модели (таблиц условных вероятностей) на сформированном наборе данных.
6. Проведено тестирование программного модуля и анализ полученных результатов, включая априорные и условные вероятности, характеризующие влияние отдельных факторов на исход событий.
7. Продемонстрированы возможности использования обученной модели для логического вывода и ответа на вопросы типа "что, если?".

**Объект исследования:** Авиационные происшествия и инциденты с самолетами гражданской авиации РФ.

**Предмет исследования:** Причинно-следственные связи между факторами риска (человеческими, техническими, организационными, средовыми, контекстуальными) и тяжестью исхода авиационных событий.

**Методы исследования:** Анализ нормативной документации и отчетов о расследовании АП, системный анализ, теория вероятностей, методы машинного обучения, аппарат Байесовских сетей доверия, программирование на Python с использованием библиотек pandas и pgmpy.

**Научная новизна:** В работе предложена и апробирована комплексная модель на основе БСД, учитывающая широкий спектр факторов различных категорий и их взаимосвязи, для количественного анализа рисков в гражданской авиации РФ. Разработан программный модуль, автоматизирующий процесс построения, обучения и первичного анализа модели на основе реальных данных об АП.

**Практическая значимость:** Разработанная модель и программный модуль могут быть использованы авиационными властями, авиакомпаниями и учебными центрами в рамках систем управления безопасностью полетов для:

* Более глубокого понимания причин АП и инцидентов.
* Выявления наиболее критичных факторов риска и их комбинаций.
* Количественной оценки эффективности различных мер по повышению безопасности.
* Поддержки принятия решений при управлении рисками.
* Совершенствования программ подготовки авиационного персонала.

**ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

* 1. **Понятие и виды авиационных происшествий**

Обеспечение безопасности в сфере воздушных перевозок требует четкого определения и классификации событий, нарушающих штатную эксплуатацию авиационной техники. Ключевым документом, устанавливающим международные стандарты в этой области, является Приложение 13 к Конвенции о международной гражданской авиации (ИКАО) [1]. В соответствии с ним, *авиационное происшествие* трактуется как событие, хронологически охватывающее период с момента посадки первого лица на борт воздушного судна с целью совершения полета и до момента выхода последнего такого лица, характеризующееся одним из следующих признаков:

* Любое лицо получает телесные повреждения со смертельным исходом или серьезные телесные повреждения в результате:
  + нахождения в данном воздушном судне;
  + непосредственного контакта с какой-либо частью воздушного судна, включая части, отделившиеся от данного воздушного судна;
  + воздействия струи газов из реактивного двигателя;
* Воздушное судно получает повреждение или происходит разрушение его конструкции, в результате чего:
  + нарушается прочность конструкции, ухудшаются технические или летные характеристики воздушного судна;
  + обычно требуется крупный ремонт или замена поврежденного элемента;
* Воздушное судно пропадает без вести или становится полностью недоступным. (ВС считается пропавшим, если официальные поиски завершены без установления его местонахождения).

В нормативно-правовом поле Российской Федерации, в частности, Постановлением Правительства РФ № 609 [2], установлена следующая классификация авиационных происшествий по тяжести последствий:

* **Катастрофа:** АП, итогом которого стала гибель хотя бы одного человека из числа находившихся на борту.
* **Авария:** АП, не приведшее к человеческим жертвам, однако повлекшее за собой повреждения воздушного судна, соответствующие критериям АП.

Наряду с авиационными происшествиями, нормативные документы выделяют **авиационные инциденты**. Данный термин охватывает события, связанные с эксплуатацией ВС, которые не классифицируются как АП, но при этом создают или могли бы создать угрозу безопасности полетов. Иными словами, инцидент – это событие, которое при несколько ином стечении обстоятельств могло бы перерасти в аварию или катастрофу. Инциденты, в свою очередь, также классифицируются по степени риска(инцидент/серьезный инцидент).  Своевременное выявление **серьезных инцидентов** (характеризующихся высокой вероятностью перерастания в АП), является неотъемлемой частью проактивного подхода к управлению безопасностью, так как позволяет идентифицировать латентные угрозы и предпринять корректирующие действия до возникновения происшествия. Примеры инцидентов включают: опасное сближение воздушных судов, выкатывание за пределы взлетно-посадочной полосы без значительных повреждений ВС, ошибки в пилотировании или УВД, создавшие угрозу безопасности, отказы критически важных систем, не приведшие к АП, посадка или взлет при погоде ниже установленного минимума и т.д.

Для целей системного анализа и статистического учета авиационные события могут также классифицироваться по ряду дополнительных признаков, включая:

* Тип задействованного воздушного судна (самолеты различных классов, вертолеты, беспилотные авиационные системы и др.).
* Этап выполнения полета, на котором произошло событие (руление, взлет, крейсерский полет, заход на посадку, посадка, пробег и т.д.).
* Установленные или предполагаемые причинные факторы (ошибки персонала – человеческий фактор, отказы авиационной техники, неблагоприятные метеорологические условия, воздействие внешних факторов, организационные недостатки и др.).

Глубокое понимание установленных определений и принципов классификации авиационных происшествий и инцидентов закладывает необходимую основу для проведения объективного расследования, всестороннего анализа факторов риска и разработки научно обоснованных мер по их предотвращению и повышению общего уровня безопасности полетов.

**1.2 Анализ статистики авиационных происшествий**

Анализ статистики авиационных происшествий играет ключевую роль в обеспечении безопасности полетов. Статистические данные позволяют выявлять тенденции, определять наиболее уязвимые области и разрабатывать целенаправленные меры по предотвращению АП.

Бюро авиационных происшествий предоставляет глобальную статистику по авиационным происшествиям [7]. Анализ этих данных позволяет увидеть общую динамику безопасности полетов, распределении АП по регионам мира, типам воздушных судов и причинам. Согласно данным BAAA, в период с 2010 по 2020 год произошло более 1500 авиационных происшествий, в результате которых погибло более 10 000 человек. Наиболее частыми причинами АП являются:

* Человеческий фактор (60% случаев);
* Технические неисправности (20%);
* Метеорологические условия (15%);
* Организационные ошибки (5%).

Вставить диаграммыНаиболее опасными этапами полета являются **взлет** и **посадка**(70% всех происшествий). За период 2012-2023 гг зафиксировано снижение общего количества авиационных происшествий на 18%, что связано с внедрением усовершенствованных систем управления рисками, цифровых технологий мониторинга и усилением международных стандартов. При этом доля катастроф сократилась на 27%, количество аварий и инцидентов без значительных последствий уменьшилось на 12%. Несмотря на позитивную динамику, человеческий фактор остается основной причиной происшествий, составляя 68% от общего числа случаев. Технические неисправности занимают 22%, а внешние условия (погодные явления, рельеф) — 8%. Организационные ошибки(недостатки в планировании или контроле) фиксируются в 2% случаев.

Вставить графикГеографический анализ демонстрирует неравномерное распределение происшествий. Наибольшее количество инцидентов зарегистрировано в регионах с высокой интенсивностью воздушного движения (Северная Америка, Азия), что объясняется высокой плотностью авиационных маршрутов. При этом в развивающихся странах доля происшествий, связанных с нарушением процедур технического обслуживания, в 1,5 раза выше, чем в государствах с развитой авиационной инфраструктурой.

Сезонные колебания также оказывают влияние: 34% происшествий, связанных с погодными условиями, приходятся на зимний период (декабрь–февраль), это связано с ростом случаев обледенения и сложных метеоявлений. В летние месяцы (июнь–август) увеличение числа инцидентов происходит из-за человеческого фактора, это может быть связано с повышенной усталостью экипажа в условиях высокого пассажиропотока.

Особого внимания заслуживает анализ влияния новых технологий. Внедрение систем предотвращения столкновений и автоматизированного управления снизило количество ошибок, связанных с восприятием и принятием решений, на 41%. Однако рост зависимости от автоматизации привел к новым рискам: 15% инцидентов в 2020–2023 гг. были вызваны некорректным взаимодействием экипажа с бортовыми системами.

В России анализ статистики АП проводится уполномоченными органами, такими как Росавиация и Межгосударственный авиационный комитет. Данные, полученные в результате анализа, используются для:

* Оценки эффективности действующих мер по обеспечению безопасности полетов.
* Выявления проблемных областей в авиационной отрасли.
* Разработки новых нормативных актов и рекомендаций.
* Определения приоритетных направлений научных исследований.

Анализ статистики АП включает в себя:

* Анализ частоты и тяжести АП: Оценка количества АП и их последствий (количество погибших, серьезно раненых, повреждения ВС).
* Анализ причин АП: Определение основных факторов, способствовавших возникновению АП (человеческий фактор, технические неисправности, метеорологические условия, организационные недостатки).
* Анализ этапов полета: Определение наиболее опасных этапов полета (взлет, посадка, набор высоты, снижение).
* Анализ типов воздушных судов: Определение типов ВС, наиболее подверженных АП.
* Анализ географического распределения АП: Определение регионов с повышенным риском АП.

Результаты статистического анализа АП используются для принятия решений на различных уровнях, от разработки национальных программ по безопасности полетов до внедрения конкретных мер по предотвращению АП в авиакомпаниях.

**1.3 Организационно-правовое обеспечение процесса анализа авиационных происшествий**

Организационно-правовое обеспечение процесса анализа авиационных происшествий представляет собой комплекс мер, направленных на регламентацию деятельности органов и лиц, участвующих в расследовании и анализе АП, а также на обеспечение их полномочий и ответственности.

Основными нормативными документами, регулирующими процесс расследования и анализа АП в Российской Федерации, являются:

* Воздушный кодекс Российской Федерации
* Постановление Правительства РФ № 609 [2]
* Федеральные авиационные правила
* Приложение 13 к Конвенции о международной гражданской авиации (ИКАО) [1].

В соответствии с этими документами, расследование АП проводится специально сформированными комиссиями, в состав которых входят представители:

* Росавиации
* Межгосударственного авиационного комитета
* Авиакомпаний
* Органов прокуратуры
* Организаций, занимающихся разработкой и производством авиационной техники.

Комиссия по расследованию АП имеет право:

* Осматривать место происшествия
* Изымать и исследовать обломки ВС
* Получать объяснения от свидетелей и участников АП
* Проводить экспертизы
* Запрашивать информацию у различных организаций.

Результаты расследования АП оформляются в виде отчета, который содержит:

* Описание обстоятельств АП
* Анализ причин АП
* Рекомендации по предотвращению подобных АП в будущем.

Организационно-правовое обеспечение также включает в себя систему ответственности за нарушение правил безопасности полетов. Лица, виновные в возникновении АП, могут быть привлечены к административной, уголовной или гражданской ответственности.

**1.4 Системы поддержки принятия решений при анализе авиационных происшествий**

Системы поддержки принятия решений играют важную роль в процессе анализа авиационных происшествий, предоставляя инструменты для сбора, обработки, анализа и визуализации данных, а также для моделирования различных сценариев развития событий.

Целью использования СППР в анализе АП является:

* Повышение эффективности и скорости расследования.
* Улучшение качества анализа причин АП.
* Обеспечение объективности и беспристрастности расследования.
* Предотвращение будущих АП.

СППР для анализа АП могут включать в себя следующие компоненты:

* **Базы данных:** Содержат информацию об АП, инцидентах, характеристиках ВС, метеорологических условиях, данных бортовых самописцев и другой релевантной информации.
* **Инструменты для обработки и анализа данных:** Позволяют проводить статистический анализ, выявлять закономерности и зависимости, моделировать развитие событий.
* **Инструменты для визуализации данных:** Предоставляют возможность наглядно представлять результаты анализа в виде графиков, диаграмм, карт и трехмерных моделей.
* **Экспертные системы:** Содержат знания экспертов в области авиационной безопасности и позволяют проводить консультации и оценки рисков.
* **Модули моделирования:** Позволяют моделировать различные сценарии развития событий и оценивать влияние различных факторов на безопасность полетов.

В настоящее время разрабатываются и внедряются СППР, основанные на технологиях искусственного интеллекта и машинного обучения. Эти системы способны автоматически выявлять аномалии в данных, прогнозировать возникновение АП и предлагать рекомендации по их предотвращению.

**1.5 Постановка задачи**

Анализ авиационных происшествий является сложной и многофакторной задачей, требующей учета большого количества взаимосвязанных факторов. Традиционные методы анализа, основанные на экспертных оценках и статистических данных, часто оказываются недостаточно эффективными для выявления скрытых причин и предотвращения будущих АП, так как носят описательный характер и направлены на выявление уникальных обстоятельств каждого случая *вручную*.

В связи с этим, актуальной задачей является разработка и применение методов моделирования причинно-следственных связей в задаче анализа авиационных происшествий. Моделирование причинно-следственных связей позволит:

* Выявлять сложные взаимосвязи между различными факторами, способствующими возникновению АП.
* Оценивать влияние различных факторов на вероятность возникновения АП.
* Прогнозировать возникновение АП на основе анализа имеющихся данных.
* Разрабатывать эффективные меры по предотвращению АП.

В рамках данной дипломной работы будет исследована возможность применения метода моделирования причинно-следственных связей, такая как Байесовские Сети Доверия. Будет разработана модель причинно-следственных связей для АП и проведена оценка ее эффективности.

Результаты работы могут быть использованы для разработки систем поддержки принятия решений для анализа авиационных происшествий, а также для повышения эффективности работы комиссий по расследованию АП.

В заключение, необходимо отметить, что разработка и внедрение эффективных методов анализа авиационных происшествий является важной задачей, направленной на повышение безопасности полетов и сохранение человеческих жизней.

**Раздел 2. МОДЕЛИ И АЛГОРИТМЫ УСТАНОВЛЕНИЯ ПРИЧИННО-  
СЛЕДСТВЕННЫХ СВЯЗЕЙ ПРИ АНАЛИЗЕ АВИАЦИОННЫХ  
ПРОИСШЕСТВИЙ**

**2.1 Гипотеза моделирования**

В основе данной работы лежит гипотеза о том, что авиационные происшествия являются результатом сложного взаимодействия множества факторов, относящихся к различным компонентам авиационной системы: человеческому фактору, техническому состоянию воздушного судна, организационным аспектам деятельности эксплуатанта и регулирующих органов, а также условиям внешней среды. Предполагается, что между этими факторами существуют причинно-следственные связи, и их совместное проявление влияет на вероятность и серьезность исхода авиационного события (инцидент, авария, катастрофа).

Для количественной оценки этих взаимосвязей и анализа влияния различных факторов на безопасность полетов предлагается использовать аппарат Дискретных Байесовских Сетей Доверия. Он отражает предполагаемые знания о причинно-следственных связях, а параметры (таблицы условных вероятностей) обучены на реальных данных об АП и инцидентах, способна адекватно моделировать риски и может быть использована для:

* Выявления наиболее критичных факторов риска.
* Оценки влияния комбинаций факторов на вероятность неблагоприятного исхода.
* Поддержки принятия решений при разработке мер по повышению безопасности полетов.

**2.2 Формализация и классификация переменных**

Создание полной модели БСД для анализа  авиационного происшествия — это чрезвычайно сложная задача, так как каждое АП уникально. Однако можно составить максимально исчерпывающий, но все же обобщенный список категорий и конкретных узлов (переменных), которые *потенциально* должны быть учтены при построении такой сети для глубокого анализа.  На основе анализа литературы и доступных отчетов об АП был сформирован набор из 43 фактора риска и 1 узла исхода. Факторы были классифицированы по категориям согласно общепринятым моделям анализа безопасности полетов (с некоторой адаптацией):

1. HF (Human Factor) – Человеческий фактор: Ошибки и нарушения, связанные с действиями летного экипажа, персонала УВД, технического обслуживания, наземного персонала. Включает ошибки навыков, принятия решений, восприятия, нарушения процедур, влияние усталости, стресса, недостатка знаний/опыта, качество CRM и т.д.
   * *Пример узла:* HF\_DecisionError (Ошибка принятия решений пилота)
   * *Состояния:* 0 – Нет, 1 – Да.
2. Tech (Technical Factor) – Технический фактор: Отказы систем ВС, конструктивные и производственные дефекты, пожар на борту.
   * *Пример узла:* Tech\_SystemFailure (Отказ системы ВС)
   * *Состояния:* 0 – Исправна, 1 – Отказ.
3. Env (Environment Factor) – Фактор внешней среды: Метеорологические условия (обледенение, турбулентность, видимость, ВНГО, ветер, гроза), состояние ВПП, состояние аэродромного оборудования (ССО, НАВ средства), рельеф местности, плотность движения, риск столкновения с птицами и т.д.
   * *Пример узла:* Env\_Icing (Обледенение)
   * *Состояния:* 0 – Отсутствует, 1 – Слабое, 2 – Умеренное, 3 – Сильное.
4. **Org (Organizational Factor) – Организационный фактор:** Качество процедур и документации, эффективность системы управления безопасностью полетов, культура безопасности, планирование и расписание, достаточность ресурсов, производственное давление, качество подготовки персонала, организация ТО, качество государственного надзора.
   * *Пример узла:* Org\_SMS\_Effectiveness (Эффективность СУБП)
   * *Состояния:* 0 – Высокая, 1 – Средняя, 2 – Низкая.
5. **Ctx (Contextual Factor) – Контекстный фактор:** Характеристики конкретной задачи или этапа полета, влияющие на риски (сложность задачи, рабочая нагрузка, этап полета, неожиданные события).
   * *Пример узла:* Ctx\_FlightPhase (Этап полета)
   * *Состояния:* 0 – Стоянка, ..., 8 – Пробег.
6. **Outcome (Исход):** Узел, представляющий серьезность авиационного события.
   * *Узел:* Outcome\_Severity (Серьезность исхода)
   * *Состояния:* 0 – Инцидент/Без последствий, 1 – Авария, 2 –Катастрофа.

Полный перечень узлов, их категорий, имен и состояний с соответствующими числовыми кодами определен в конфигурационном файле bn\_config.json (см. Приложение Y). Каждому состоянию присвоен неотрицательный целочисленный код, начиная с 0.

**2.3 Математическая модель анализа авиационных происшествий**

**2.3.1 Выбор подхода к моделированию**

Анализ авиационных происшествий отличается наличием множества взаимосвязанных факторов различной природы (человеческие, технические, средовые, организационные) и неопределенностью их влияния на конечный исход (инцидент, авария, катастрофа). Необходимо было выбрать математический аппарат, способный адекватно отразить эти особенности. Рассматривались следующие подходы:

1. **Детерминированные модели (например, деревья событий/отказов):** Позволяют наблюдать последовательности событий, ведущие к АП. Однако они плохо справляются с неопределенностью, вероятностным характером влияния факторов и сложными взаимозависимостями между ними.
2. **Статистические модели (регрессионный анализ, корреляционный анализ):** Позволяют выявить статистические связи между факторами и исходом на основе больших объемов данных. Недостатком является сложность интерпретации причинно-следственных связей (корреляция не всегда означает причинность) и трудность учета сложных нелинейных взаимодействий и априорных знаний.
3. **Модели на основе теории графов и искусственного интеллекта:**
   * *Экспертные системы:* Основаны на правилах "если-то", полученных от экспертов. Эффективны в узких областях, но сложны в разработке и поддержке, подвержены субъективности.
   * *Нейронные сети:* Мощный инструмент для выявления скрытых закономерностей в данных, но являются "черным ящиком", затрудняя интерпретацию причинно-следственных связей, что критично для анализа безопасности.
   * *Байесовские Сети Доверия (БСД):* Сочетают вероятностный подход с наглядным графическим представлением зависимостей. Позволяют интегрировать как статистические данные, так и экспертные знания, моделировать неопределенность и выполнять логический вывод. Структура сети явно отображает предполагаемые причинно-следственные связи.

Учитывая необходимость моделирования неопределенности, представления причинно-следственных связей и возможность обучения на ограниченном объеме реальных данных об АП, в качестве основного математического аппарата были выбраны Дискретные Байесовские Сети Доверия.

**Байесовская сеть**

Вероятностные модели позволяют нам использовать вероятностный вывод для вычисления распределения вероятностей для набора ненаблюдаемых («гипотетических») переменных при заданном наборе наблюдаемых переменных.

Ориентированный граф, каждый узел которого аннотирован количественной вероятностной информацией:

1. Каждый узел соответствует случайной величине, непрерывной или дискретной
2. Набор направленных связей или стрелок соединяет пары узлов. Если стрелка ведёт от узла X к узлу Y, то X считается родительским узлом для Y. В графе нет направленных циклов (DAG).
3. Каждый узел  имеет условное распределение вероятностей *P(|Parents(Xi)),* это количественно определяет влияние родителей на узел.
4. Направленная дуга от A к B означает, что A «оказывает прямое влияние на» или «является причиной» B.

Свидетельство в пользу A увеличивает правдоподобие B (дедуктивное влияние от причин к следствиям).

Свидетельство в пользу B увеличивает правдоподобие A (абдуктивное влияние от следствий к причинам).

1. Кодируют предположения об условной независимости.

Байесовские сети можно рассматривать как кодировку набора утверждений об условной независимости.

*P(,..., )=*

**2.3.2 Построение Структуры БСД**

Ключевым этапом является определение структуры сети – направленного ациклического графа, где узлы – это переменные (факторы и исход), а ребра – прямые зависимости. Поскольку алгоритмы автоматического обучения структуры БСД из данных требуют очень больших выборок, которых в области редких событий, как АП, обычно нет, структура сети в данной работе строилась на основе **экспертного подхода**:

1. Идентификация узлов: Определен перечень релевантных переменных (43 фактор + 1 исход) на основе анализа нормативной документации (ПРАПИ, ФАП), моделей безопасности (SHELL, HFACS), отчетов МАК и Росавиации, а также файла конфигурации bn\_config.json (см. п. 2.2).

Узлы: Факторы риска (`HF\_\*`, `Tech\_\*`, `Env\_\*`, `Org\_\*`, `Ctx\_\*`) и Исход (`Outcome\_Severity`).

1. Установление связей: На основе логики развития авиационных событий и общепринятых знаний в области безопасности полетов были установлены направленные ребра между узлами. При этом учитывалась иерархия факторов:
   * *Коренные причины (Root Causes):* Часто организационные факторы (Org\_\*) и базовые условия среды (Env\_Terrain) рассматривались как узлы без родителей в модели.
   * *Промежуточные факторы:* Контекстные (Ctx\_\*), средовые (Env\_\*), технические (Tech\_\*) и человеческие (HF\_\*) факторы часто зависят от коренных причин и друг от друга. Например, погода (Env\_\*) влияет на сложность (Ctx\_TaskComplexity) и рабочую нагрузку (Ctx\_Workload), что, в свою очередь, влияет на ошибки пилота (HF\_\*). Некачественное ТО (HF\_Maint\_Quality - *в итоговой модели удален, но был в первоначальной*) может привести к отказу техники (Tech\_SystemFailure).
   * *Непосредственные причины:* Ошибки персонала (HF\_\*), отказы техники (Tech\_\*), экстремальные условия среды (Env\_\*) напрямую влияют на исход (Outcome\_Severity).
2. Проверка ацикличности: Построенная структура была проверена на отсутствие циклов, что является обязательным условием для БСД.

В результате была получена сложная сеть зависимостей (81 ребра для 43 узла), представленная графически в Приложении X, которая, по предположению, более адекватно отражает реальность, чем упрощенная Наивная Байесовская модель.

**2.3.3 Определение Параметров БСД (Обучение CPT)**

Рассмотрим упрощенный случай с 3 факторами:

1. `Env\_Visibility` (Видимость): {0: Выше мин., 1: Ниже мин.}

2. `HF\_SkillError` (Ошибка навыков): {0: Нет, 1: Да}

3. `Outcome\_Severity` (Исход): {0: Инцидент, 1: Авария, 2: Катастрофа}

Чтобы задать ПСРВ, нужно перечислить \*\*все\*\* 2 \* 2 \* 3 = 12 комбинаций и присвоить вероятность каждой:

|  |  |  |  |
| --- | --- | --- | --- |
| Видимость | ОшибкаНав | Исход | Вероятн. |
| 0 | 0 | 0 | 0,60 |
| 0 | 0 | 1 | 0,02 |
| 0 | 0 | 2 | 0,01 |
| 0 | 1 | 0 | 0,10 |
| 0 | 1 | 1 | 0,03 |
| 0 | 1 | 2 | 0,02 |
| 1 | 0 | 0 | 0,15 |
| 1 | 0 | 1 | 0,01 |
| 1 | 0 | 2 | 0,005 |
| 1 | 1 | 0 | 0,02 |
| 1 | 1 | 1 | 0,015 |
| 1 | 1 | 2 | 0,02 |
| Сумма: 1,00 | | | |

Проблема: С ростом числа факторов размер таблицы становится астрономическим!

**Маргинализация и Условные Вероятности из ПСРВ**

Имея ПСРВ, можно вычислить любую вероятность.

Маргинализация (расчет априорной вероятности):

P(ОшибкаНав = 1) = Сумма вероятностей всех строк, где ОшибкаНав = 1

= 0.10 + 0.03 + 0.02 + 0.02 + 0.015 + 0.02 = 0.205

Условная вероятность (вывод):

P(Исход=Катастрофа | Видимость=1) = P(Исход=2, Видимость=1) / P(Видимость=1)

P(Исход=2, Видимость=1) = P(В=1, ОН=0, И=2) + P(В=1, ОН=1, И=2) = 0.005 + 0.02 = 0.025

P(Видимость=1) = Сумма вероятностей всех строк, где Видимость=1 = 0.15 + 0.01 + 0.005 + 0.02 + 0.015 + 0.02 = 0.22

P(Исход=2 | Видимость=1) = 0.025 / 0.22 ≈ 0.114

**Параметры (CPT)**

Для каждого узла хранится Таблица Условных Вероятностей (CPT).

CPT показывает вероятность каждого состояния узла при всех комбинациях состояний его *прямых родителей*.

Пример: CPT для HF\_SkillError может зависеть от HF\_Fatigue, HF\_Stress, Ctx\_Workload. Таблица будет содержать вероятности `P(HF\_SkillError=1 | HF\_Fatigue=..., HF\_Stress=..., Ctx\_Workload=...)` для всех комбинаций состояний родителей.

Для *корневых узлов* (без родителей, например, `Org\_RegulatoryOversight` в вашей модели) CPT содержит *априорные вероятности* их состояний

БСД компактно кодирует ПСРВ за счет использования условных независимостей, заложенных в структуре графа.

ПСРВ: Требует вероятностей (где N ~ 62 узлов, k ~ 2-4 состояния). Огромное число!

**Условная Независимость в БСД**

Концепция: Знание состояния одной переменной может сделать другую переменную независимой от третьей.

Пример: Org\_TrainingQuality –> HF\_KnowledgeSkill –> HF\_SkillError

Без знания HF\_KnowledgeSkill : Низкое качество обучения (Org\_TrainingQuality=1) повышает вероятность ошибки навыков (`HF\_SkillError=1`). Они зависимы.

При известном HF\_KnowledgeSkill : Если мы точно знаем, что у пилота недостаточные знания (HF\_KnowledgeSkill=1), то информация о качестве его первоначального обучения (Org\_TrainingQuality) уже *не добавляет* информации к вероятности ошибки навыков (HF\_SkillError). То есть, HF\_SkillError *условно независимо* от Org\_TrainingQuality при заданном `HF\_KnowledgeSkill.

Марковское Покрытие : узел условно независим от всех остальных узлов сети при условии знания состояний его родителей, детей и других родителей его детей.

**Интерпретация Связей в БСД**

1. Причинная Цепь: Фактор А -> Фактор Б -> Фактор В

Пример: Org\_Pressure -> HF\_Stress -> HF\_DecisionError (Давление -> Стресс -> Ошибка Решения).

Org\_Pressure и HF\_DecisionError зависимы, но становятся независимыми, если состояние HF\_Stress *известно*.

2. Общая Причина: Фактор А -> Фактор Б, Фактор А -> Фактор В

Пример: Env\_Icing -> Ctx\_TaskComplexity, Env\_Icing -> Tech\_SystemFailure (Обледенение -> Сложность Задачи; Обледенение -> Отказ Системы).

Ctx\_TaskComplexity и Tech\_SystemFailure *зависимы* (т.к. у них общая причина), но становятся *независимыми*, если состояние Env\_Icing известно.

3. Общее Следствие (Коллайдер): Фактор А -> Фактор В <- Фактор Б

Пример: HF\_SkillError -> Outcome\_Severity <- Tech\_SystemFailure (Ошибка Пилота -> Исход <- Отказ Техники).

HF\_SkillError иTech\_SystemFailure` априорно *независимы*.

НО: Если известен исход (Outcome\_Severity, например, произошла авария), то факторы HF\_SkillError и Tech\_SystemFailure становятся *зависимыми* (эффект "объяснения прочь": если мы знаем, что была ошибка пилота, это снижает нашу уверенность, что причиной был отказ техники, и наоборот).

**Обучение CPT**

После определения структуры необходимо количественно описать силу связей – определить Таблицы Условных Вероятностей (CPT) *P(| Parents())* для каждого узла . Это было сделано путем обучения модели на данных:

1. Источник данных: Использовался подготовленный и уточненный набор данных из 146 АП и инцидентов (aviation\_incidents\_refined.csv), где каждому событию соответствуют числовые коды состояний для всех узлов модели.
2. Метод оценки: Применен Байесовский метод оценки (BayesianEstimator из библиотеки pgmpy). Этот метод выбран по нескольким причинам:
   * Работа с малыми выборками: Данные об АП (особенно о катастрофах и авариях) относительно редки. Байесовская оценка, в отличие от оценки максимального правдоподобия, позволяет избежать нулевых вероятностей для комбинаций состояний, не встретившихся в обучающей выборке, за счет использования априорных знаний (псевдоподсчетов).
   * Сглаживание**:** Параметр equivalent\_sample\_size (установлен в 10) определяет "вес" априорного равномерного распределения, добавляемого к наблюдаемым частотам, что делает оценки более робастными.
3. **Процесс обучения:** Для каждого узла  и каждой комбинации состояний его родителей Parents() = j библиотека pgmpy подсчитывает частоты  (сколько раз  был в состоянии k при родителях j) и  (сколько раз родители были в состоянии j) в данных data\_to\_fit. Затем, с учетом псевдоподсчетов (α), рассчитывается условная вероятность по формуле, аппроксимирующей Байесовскую оценку с априорным распределением Дирихле:  
   P(= k | Parents() = j) ≈ ( +) / ( + )
4. **Результат:** Набор CPT для всех узлов модели, количественно описывающий вероятностные зависимости между факторами и исходом АП на основе проанализированных данных.

**2.3.4 Механизм Логического Вывода**

Обученная БСД позволяет выполнять вероятностный логический вывод, т.е. отвечать на вопросы вида "Какова вероятность события A при условии, что произошло событие B?". В данной работе используется алгоритм Исключения Переменных (VariableElimination):

1. Вход: Обученная БСД (структура + CPT), интересующие переменные (variables, например, [outcome\_node]) и известные факты (evidence, например, {Env\_Visibility: 1, HF\_SkillError: 1}).
2. Процесс: Алгоритм использует структуру сети и CPT для эффективного вычисления апостериорного распределения вероятностей для интересующих переменных путем последовательного перемножения факторов (CPT) и суммирования (маргинализации) по переменным, не входящим в запрос или свидетельства.
3. Выход: Новое вероятностное распределение для интересующих переменных, обновленное с учетом свидетельств (P(variables | evidence)).

Этот механизм позволяет использовать модель для прогнозирования, диагностики и сценарного анализа.

Таким образом, разработанная математическая модель представляет собой Дискретную Байесовскую Сеть Доверия с реалистичной структурой, параметры которой обучены на исторических данных об АП с использованием Байесовской оценки, и которая позволяет выполнять вероятностный логический вывод для анализа причинно-следственных связей и оценки рисков.

**2.4 Алгоритм установления причинно-следственных связей**

После построения и обучения математической модели БСД (как описано в п. 2.3) необходимо было разработать и применить алгоритм, позволяющий не только визуализировать предполагаемые зависимости (заложенные в структуре графа), но и количественно оценить силу и характер влияния различных факторов на исход авиационного происшествия. Этот алгоритм основан на анализе параметров обученной сети (CPT) и использовании механизма вероятностного логического вывода.

Этапы разработанного алгоритма:

1. Идентификация Потенциальных Связей (Анализ Структуры):
   * Действие: Программный анализ структуры построенного DAG (model.edges()). Наличие направленного ребра от узла фактора  к узлу (другому фактору или исходу Y) интерпретируется как гипотеза о прямой причинно-следственной связи, заложенная на этапе проектирования модели.
   * Результат: Формирование списка предполагаемых прямых зависимостей, подлежащих количественной оценке. Визуализация графа (см. Приложение X) служит для наглядного представления этих гипотез.
2. Количественная Оценка Силы Прямых Связей (Анализ CPT):
   * Действие: Для каждого узла , имеющего родителей *Parents()*, из обученной модели извлекается соответствующая таблица условных вероятностей *P( | Parents()).*
   * Анализ: Анализируется, как изменяется распределение вероятностей состояний узла  при изменении состояний его прямых родителей Parents().
   * Пример: Для CPT узла Outcome\_Severity анализируется, насколько сильно вероятность аварии P(Outcome\_Severity=1 | ...) или катастрофы P(Outcome\_Severity=2 | ...) зависит от того, находится ли его прямой родитель HF\_SkillError в состоянии 0 ("Нет ошибки") или 1 ("Ошибка") при фиксированных состояниях других родителей. Значительное изменение вероятности указывает на сильную прямую связь.
   * Результат: Получение количественных оценок силы прямых зависимостей, заложенных в структуре сети.
3. Оценка Общего Влияния Факторов на Исход (Анализ Чувствительности и Ранжирование):
   * Цель: Определить, какие факторы оказывают наибольшее *общее* (прямое и косвенное) влияние на вероятность неблагоприятного исхода (аварии или катастрофы) в определенном контексте.
   * Разработка Метрики Влияния**:** В качестве метрики влияния был выбран *коэффициент увеличения риска* – отношение условной вероятности неблагоприятного исхода при наличии "проблемного" состояния фактора к базовой вероятности неблагоприятного исхода.
   * Алгоритм:  
     1.  Определение базового сценария: Задается набор свидетельств E\_{base} (из файла incident\_evidence.json), описывающий интересующий контекст (например, погодные условия, тип операции). Если свидетельств нет, используется априорное распределение.  
     2.  Расчет базовой вероятности: С помощью механизма логического вывода (VariableElimination.query) рассчитывается базовая вероятность неблагоприятного исхода P\_{base} = P(Outcome\_Severity > 0 | E\_{base}).  
     3.  Итерация по факторам: Для каждого фактора риска  (из списка узлов модели, кроме исхода):  
     – Определяется его "проблемное" состояние j (первое состояние с кодом > 0).  
     – Формируется новый набор свидетельств  = E\_{base} с принудительным установлением   = j.  
     – Рассчитывается условная вероятность неблагоприятного исхода при данном факторе:  = P(Outcome\_Severity > 0 |  ).  
     – Рассчитывается коэффициент влияния для фактора  :    =   / P\_{base} (с обработкой деления на ноль).  
     4. Ранжирование**:** Факторы   сортируются по убыванию .
   * Реализация: Данный алгоритм реализован в функции assess\_factor\_influence скрипта predict\_outcome.py.
   * Результат: Ранжированный список факторов, упорядоченный по степени их влияния на повышение риска неблагоприятного исхода в заданном контексте. Это позволяет выявить наиболее критичные факторы для конкретных условий.
4. Анализ Комбинаций Факторов (Сценарный Анализ):
   * Действие: Использование механизма логического вывода (VariableElimination.query) для расчета вероятности исхода P(Outcome\_Severity | E\_{scenario}) при задании различных комбинаций состояний факторов в качестве свидетельств E\_{scenario}.
   * Пример: Расчет P(Outcome\_Severity=2 | Env\_Visibility=1, HF\_SkillError=1) для оценки совместного влияния плохой видимости и ошибки пилота на вероятность катастрофы.
   * Результат: Возможность количественной оценки синергетических (или компенсирующих) эффектов от взаимодействия различных факторов риска.

Заключение по алгоритму:  
Разработанный алгоритм, сочетающий анализ структуры БСД, параметров CPT и применение логического вывода для анализа чувствительности и сценариев, позволяет не только выявить гипотетические причинно-следственные связи, но и количественно оценить их силу и значимость различных факторов риска (как по отдельности, так и в комбинации) для исхода авиационных происшествий. Это обеспечивает основу для объективного анализа и приоритизации мер по повышению безопасности полетов.

**Раздел 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**3.1 Функциональные требования к программе и обоснование выбора среды программирования**

Для реализации и апробации разработанной математической модели анализа авиационных происшествий на основе Байесовских сетей доверия (БСД) было необходимо создать программное обеспечение (ПО). На этапе проектирования были сформулированы следующие функциональные требования к данному ПО:

**Функциональные требования**

1. Конфигурируемость модели:
   * ПО должно позволять загружать описание структуры узлов БСД (факторов риска и исхода) и их дискретных состояний из внешнего конфигурационного файла (формат JSON).
   * ПО должно позволять определять структуру зависимостей (направленные ребра графа) между узлами внутри программного кода или из внешнего источника (в данной реализации структура задана в коде).
2. Загрузка и предобработка данных:
   * ПО должно обеспечивать чтение исходных данных об авиационных событиях из структурированного файла (формат CSV).
   * Реализация механизмов валидации данных на соответствие конфигурации модели (проверка наличия столбцов, типов данных, допустимых кодов состояний).
   * Обеспечение предобработки данных: обработка пропущенных значений (NaN), некорректных записей (например, бесконечности), преобразование данных к целочисленному формату, требуемому для обучения дискретной БСД.
3. Построение и обучение БСД:
   * Создание объекта дискретной БСД на основе загруженной конфигурации и определенной структуры.
   * Автоматическая проверка построенной структуры графа на ацикличность.
   * Реализация механизма обучения параметров модели (расчета таблиц условных вероятностей - CPT) на основе подготовленных данных с использованием Байесовского метода оценки (BayesianEstimator).
   * Возможность сохранения обученной модели (структуры и параметров) во внешний файл для последующего использования.
4. Анализ и логический вывод:
   * Возможность загрузки ранее обученной модели из файла.
   * Расчет и вывод априорных вероятностей для узла исхода и ключевых факторов риска.
   * Расчет и вывод условных вероятностей исхода при наличии отдельных факторов риска (P(Исход | Фактор > 0)).
   * Реализация механизма вероятностного логического вывода (VariableElimination) для расчета апостериорных вероятностей исхода при задании произвольного набора свидетельств (evidence) из внешнего файла (формат JSON).
   * Реализация алгоритма оценки и ранжирования влияния факторов риска на неблагоприятные исходы на основе анализа чувствительности модели к изменению состояния факторов при заданных свидетельствах.
5. Визуализация (Опционально):
   * Возможность генерации и сохранения графического представления структуры разработанной БСД для наглядности.

**Обоснование выбора среды программирования и инструментов**

Исходя из сформулированных функциональных требований, в качестве среды разработки был выбран язык программирования *Python 3* и его экосистема библиотек для научных вычислений и анализа данных. Данный выбор обусловлен следующими факторами:

1. Наличие специализированных библиотек для БСД: Библиотека pgmpy ([https://pgmpy.org/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fpgmpy.org%2F)) является мощным и активно развивающимся инструментом с открытым исходным кодом, специально предназначенным для работы с вероятностными графическими моделями, включая Байесовские сети. Она предоставляет классы для:
   * Определения структуры модели (DiscreteBayesianNetwork).
   * Обучения параметров различными методами (BayesianEstimator, MaximumLikelihoodEstimator).
   * Выполнения точного и приближенного логического вывода (VariableElimination, BeliefPropagation и др.).
   * Анализа структуры и параметров модели.  
     Использование pgmpy позволяет реализовать ключевые требования к моделированию и анализу БСД без необходимости разработки сложных вероятностных алгоритмов с нуля.
2. Эффективная работа с данными: Библиотека pandas является стандартом для обработки и анализа табличных данных в Python. Она обеспечивает удобные и производительные инструменты для:
   * Чтения данных из различных форматов (включая CSV).
   * Очистки, преобразования, фильтрации данных.
   * Обработки пропущенных значений.
   * Расчета описательных статистик и частот, необходимых для анализа и подготовки данных к обучению.
3. Научные вычисления: Библиотека numpy предоставляет фундаментальные структуры данных (многомерные массивы) и функции для эффективных числовых операций, которые используются многими другими библиотеками, включая pandas и pgmpy.
4. Работа с JSON: Встроенный в Python модуль json обеспечивает простую и надежную работу с файлами конфигурации и входными данными в формате JSON.
5. Визуализация:
   * Библиотека matplotlib является стандартной библиотекой для создания статических, анимированных и интерактивных визуализаций в Python.
   * Библиотека networkx предоставляет инструменты для создания, манипулирования и изучения структуры сложных сетей (графов).
   * Взаимодействие networkx с установленной программой Graphviz (через интерфейсы pygraphviz или pydot) позволяет генерировать качественные графические представления структуры БСД, используя автоматические алгоритмы компоновки графов (например, dot).
6. Общие преимущества Python:
   * Открытый исходный код и большое сообщество: Доступность, обилие документации, примеров и готовых решений.
   * Читаемость и простота разработки: Позволяет быстрее реализовать и отладить алгоритмы.
   * Кроссплатформенность: Обеспечивает работу ПО на различных операционных системах.
   * Возможности расширения: Легкая интеграция с другими библиотеками и инструментами при необходимости.
   * Сериализация: Встроенный модуль pickle позволяет легко сохранять и загружать сложные объекты Python, такие как обученная модель pgmpy.

Таким образом, выбор Python и библиотек pgmpy, pandas, numpy, networkx, matplotlib является обоснованным, поскольку данная среда предоставляет все необходимые инструменты для эффективной реализации функциональных требований по разработке, обучению, анализу и визуализации Байесовской сети доверия для анализа авиационных происшествий.

**3.2 Создание программного продукта**

**3.2.1 Формирование Набора Исходных Данных**

Для обучения и тестирования математической модели был сформирован набор данных на основе официальных отчетов и донесений об авиационных происшествиях и инцидентах с самолетами гражданской авиации РФ за период 2012-2023 гг. Источниками послужили сводки филиалов ФГБУ «Авиаметтелеком Росгидромета», ФГБУ «УГМС», а также выписки из отчетов МАК и Росавиации, представленные в аналитических материалах.

Процесс формирования данных включал:

1. Отбор релевантных событий**:** Из отчетов были выбраны авиационные события (инциденты, серьезные инциденты, аварии, катастрофы), связанные с самолетами и имеющие отношение к метеорологическим, техническим, организационным или человеческим факторам. События, связанные исключительно с внешним вмешательством (незаконное вмешательство), или происшествия с вертолетами/БПЛА/СВС (если не оговорено иное), исключались из основной выборки для данной модели.
2. Идентификация факторов**:** На основе файла конфигурации bn\_config.json, содержащего перечень и описание релевантных факторов (узлов БСД) и их возможных состояний (с числовыми кодами), для каждого отобранного события производился анализ текстовых описаний.
3. Кодирование данных**:** Каждому событию (строке данных) присваивались числовые коды состояний для каждого фактора в соответствии с bn\_config.json. Особое внимание уделялось столбцу "Причина(ы) АП", информация из которого использовалась для уточнения и верификации кодов состояний в соответствующих столбцах-факторах.
4. Формирование таблицы: Данные были сведены в таблицу формата CSV (aviation\_incidents\_refined.csv), где строки соответствуют авиационным событиям, а столбцы – факторам (узлам БСД) и исходу (Outcome\_Severity).
5. Очистка данных: Проведена обработка пропущенных значений (заполнение нулем с соответствующим предупреждением о потенциальной некорректности) и проверка на наличие недопустимых (например, отрицательных) значений.

В результате был сформирован набор данных, содержащий 146 записей (авиационных событий), пригодный для обучения дискретной Байесовской сети.

**3.2.2 Программирование**

На языке программирования Python с использованием библиотеки pgmpy был разработан программный модуль, выполняющий следующие функции:

1. Загрузка Входных Данных:

* Описание: Первым шагом является загрузка конфигурации сети из JSON и данных об инцидентах из CSV.
* Реализация: Используются стандартные функции: json.load() для парсинга JSON-файла в словарь Python (config) и pandas.read\_csv() для чтения CSV-файла в DataFrame (df). Кодировка 'utf-8' указана для поддержки кириллицы в конфигурации. Обработка исключений FileNotFoundError и общих Exception обеспечивает робастность на этапе загрузки.

import json

import pandas as pd

CONFIG\_FILE = 'bn\_config.json'

DATA\_FILE = 'aviation\_incidents\_refined.csv'

try:

with open(CONFIG\_FILE, 'r', encoding='utf-8') as f:

config = json.load(f)

df = pd.read\_csv(DATA\_FILE)

print(f"Данные CSV ('{DATA\_FILE}', {df.shape[0]} строк) и конфигурация JSON ('{CONFIG\_FILE}') успешно загружены.")

except FileNotFoundError:

# ... обработка ошибки ...

exit()

except Exception as e:

# ... обработка ошибки ...

exit()

2. Валидация Данных и Конфигурации:

* Описание: Критически важный этап для обеспечения совместимости данных и модели. Проверяется наличие всех необходимых узлов в данных, соответствие значений в данных допустимым кодам состояний из JSON, отсутствие отрицательных кодов. Также формируется словарь state\_names, необходимый для pgmpy.
* Реализация: Реализован цикл, итерирующий по описаниям факторов и исхода в config. Для каждого узла:
  + Проверяется наличие соответствующего столбца в df
  + Извлекаются и сортируются допустимые коды состояний (allowed\_states)
  + state\_names[node\_id] = allowed\_states сохраняет эту информацию
  + Столбец данных преобразуется в числовой формат с помощью pd.to\_numeric(errors='coerce'), где errors='coerce' заменяет нечисловые значения на NaN, предотвращая падение скрипта
  + dropna().unique() получает уникальные валидные числовые значения
  + Проводится проверка каждого уникального значения на принадлежность к allowed\_states и на неотрицательность
  + Устанавливается флаг issues\_found\_data\_config при обнаружении любых несоответствий. Если флаг установлен, скрипт завершается.

# Фрагмент цикла валидации фактора

node\_id = factor.get('id')

if node\_id in df.columns:

# ... (проверки JSON) ...

allowed\_states = sorted([s['code'] for s in factor['states']])

state\_names[node\_id] = allowed\_states

nodes\_in\_data\_and\_config.append(node\_id)

numeric\_series = pd.to\_numeric(df[node\_id], errors='coerce')

unique\_vals = numeric\_series.dropna().unique()

unique\_int\_vals = [int(v) for v in unique\_vals if v == int(v)] # Целые числа

if any(val < 0 for val in unique\_int\_vals): # Проверка на < 0

print(f"ОШИБКА: Отрицательные значения в '{node\_id}'.")

issues\_found\_data\_config = True

invalid\_vals = [val for val in unique\_int\_vals if val not in allowed\_states]

if invalid\_vals:

print(f"ОШИБКА: Недопустимые значения {invalid\_vals} в '{node\_id}'. Разрешены: {allowed\_states}")

issues\_found\_data\_config = True

# ... (обработка отсутствия столбца) ...

3. Очистка и Подготовка Данных:

* Описание: Данные фильтруются (остаются только столбцы, прошедшие валидацию), преобразуются к нужному типу и обрабатываются пропуски.
* Реализация:
  + df\_prepared = df[nodes\_in\_data\_and\_config].copy(): Создается копия DataFrame только с нужными столбцами
  + Цикл применяет pd.to\_numeric(errors='coerce') к каждому столбцу df\_prepared для единообразия
  + df\_prepared.replace([np.inf, -np.inf], np.nan, inplace=True): Заменяет бесконечные значения (если вдруг появились) на NaN
  + df\_prepared.fillna(0, inplace=True): Заполняет все пропуски (NaN) нулями. Выбор нуля сделан для простоты, исходя из допущения, что пропуск чаще всего означает "нормальное" или "неприменимое" состояние (код 0). Это требует внимательной проверки для конкретных данных
  + df\_prepared = df\_prepared.astype(int): Преобразует все данные в целочисленный тип, как того требует DiscreteBayesianNetwork

nodes\_in\_data\_and\_config = list(state\_names.keys()) # Узлы, которые есть и в JSON и в данных

df\_prepared = df[nodes\_in\_data\_and\_config].copy()

for col in nodes\_in\_data\_and\_config:

df\_prepared[col] = pd.to\_numeric(df\_prepared[col], errors='coerce')

df\_prepared.replace([np.inf, -np.inf], np.nan, inplace=True)

df\_prepared.fillna(0, inplace=True) # ВАЖНОЕ ДОПУЩЕНИЕ

df\_prepared = df\_prepared.astype(int)

4. Определение Структуры БСД:

* Описание**:** Создается граф зависимостей между узлами.
* Реализация:
  + Список edges заполняется кортежами ('родитель', 'ребенок') на основе экспертных знаний о причинно-следственных связях (блок edges.extend([...]))
  + unique\_edges = list(set(edges)) удаляет возможные дубликаты
  + model\_nodes\_list формируется из узлов, реально присутствующих в df\_prepared
  + model = DiscreteBayesianNetwork(): Создается пустая модель БСД
  + model.add\_nodes\_from(model\_nodes\_list): Добавляются узлы
  + model.add\_edges\_from(unique\_edges): Добавляются ребра
  + nx.is\_directed\_acyclic\_graph(model): Используется библиотека networkx для проверки графа на отсутствие циклов.

edges = []

# ... (большой блок edges.extend([...]) с определением всех связей) ...

unique\_edges = list(set(edges))

model\_nodes\_list = [node for node in all\_nodes\_in\_config if node in df\_prepared.columns]

model = DiscreteBayesianNetwork()

model.add\_nodes\_from(model\_nodes\_list)

model.add\_edges\_from(unique\_edges) # pgmpy сама проверяет на DAG при добавлении ребер, но доп. проверка с nx надежнее

if not nx.is\_directed\_acyclic\_graph(model):

raise ValueError("Структура сети содержит циклы!")

print("Структура сети определена и является ацикличной.")

5. Проверка данных перед обучением:

* Описание: Финальная проверка соответствия подготовленных данных (data\_to\_fit) и ожиданий модели (кардинальность из state\_names)
* Реализация:
  + data\_to\_fit = df\_prepared[model\_nodes\_list].copy(): Создается срез данных только для узлов модели
  + Цикл по model.nodes(): Для каждого узла сравнивает максимальное/минимальное значение в data\_to\_fit с ожидаемыми границами (0 и len(state\_names[node]) - 1). Проверяет на отрицательные значения. Выводит предупреждения для узлов с одним уникальным значением
  + Если найдены ошибки, скрипт завершается
* # Фрагмент цикла проверки
* for node in model.nodes():
* # ... (проверки наличия node в state\_names и data\_to\_fit) ...
* model\_cardinality = len(state\_names[node])
* expected\_max\_state = model\_cardinality - 1
* actual\_max\_state = data\_to\_fit[node].max()
* actual\_min\_state = data\_to\_fit[node].min()
* # ... (вывод информации) ...
* if actual\_max\_state > expected\_max\_state or actual\_min\_state < 0:
* print(f" ОШИБКА: Значения для '{node}' вне диапазона [0, {expected\_max\_state}].")
* issues\_found\_cardinality = True

# ... (проверка флага issues\_found\_cardinality и выход при ошибке) ...

6. Обучение Параметров (CPT):

* Описание: Расчет таблиц условных вероятностей.
* Реализация: Используется метод model.fit().
  + data=data\_to\_fit: Передаются подготовленные данные
  + estimator=BayesianEstimator: Выбран Байесовский оценщик
  + state\_names=state\_names: Обязательно передается словарь, чтобы pgmpy знала все возможные дискретные состояния и их порядок для каждого узла
  + equivalent\_sample\_size=10: Параметр для сглаживания (априорные псевдоподсчеты)
  + Весь вызов обернут в try...except для перехвата ValueError (часто указывает на проблемы с данными/состояниями) и других исключений. В случае ValueError выводится дополнительная диагностика уникальных значений
* try:
* print("\nНачало обучения параметров...")
* model.fit(data=data\_to\_fit, estimator=BayesianEstimator, state\_names=state\_names, equivalent\_sample\_size=10)
* print("Параметры сети (CPT) успешно обучены.")
* except ValueError as e:
* print(f"\nОШИБКА ValueError во время model.fit: {e}")
* # ... (вывод диагностики уникальных значений) ...
* exit()
* except Exception as e:
* print(f"\nНеожиданная ОШИБКА во время model.fit: {e}")

exit()

7. Сохранение Модели:

* Описание: Сохранение обученной модели для использования в других модулях.
* Реализация: Используется модуль pickle. pickle.dump(model, f) сериализует объект model (включая структуру и CPT) и записывает его в бинарный файл ('wb')
* try:
* with open(MODEL\_FILE, 'wb') as f:
* pickle.dump(model, f)
* print(f"Обученная модель сохранена в файл: {MODEL\_FILE}")
* except Exception as e:

print(f"Ошибка при сохранении модели: {e}")

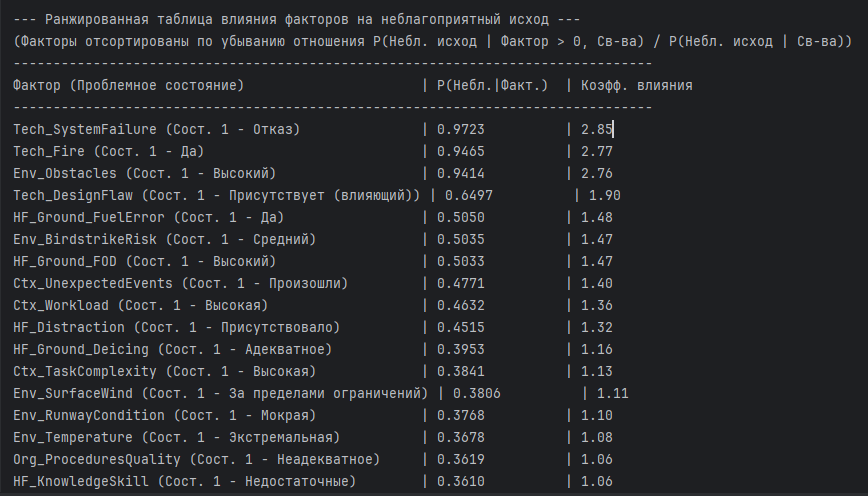
8. Анализ вероятностей и визуализация:

* Описание: Расчет и вывод базовых вероятностных показателей, логический вывод на примерах, генерация графа
* Реализация:
  + Априорные вероятности: Рассчитываются с помощью pandas.value\_counts(normalize=True) на data\_to\_fit
  + Условные вероятности: Рассчитываются путем фильтрации data\_to\_fit по условию (df\_problem = data\_to\_fit[data\_to\_fit[factor\_id].isin(problem\_codes)]) и последующего применения value\_counts(normalize=True) к отфильтрованным данным
  + Логический вывод: Создается inference = VariableElimination(model). Выполняются запросы inference.query(variables=[...], evidence={...}). Вероятности извлекаются с result\_distribution.get\_value(...)
  + Визуализация: Используется networkx.DiGraph(model.edges()) для создания графа, nx.nx\_agraph.graphviz\_layout(graph, prog='dot') для расчета позиций узлов (требует Graphviz и pygraphviz/pydot) и nx.draw(...), plt.savefig(...) для отрисовки и сохранения.

**3.2.3 Модуль Предсказания и Оценки Влияния (predict\_outcome.py)**

Данный модуль использует **результаты** работы BSD.py.

1. **Загрузка Ресурсов:**
   * **Описание:** Загрузка конфигурации JSON, обученной модели из .pkl файла и свидетельств из evidence.json.
   * **Реализация:** Используются функции load\_config, load\_model, load\_evidence, которые содержат json.load и pickle.load соответственно, обернутые в try...except для обработки ошибок. В load\_model добавлена проверка if not model.cpds, чтобы убедиться, что загруженная модель действительно обучена.
2. **Выполнение Логического Вывода:**
   * **Описание:** Расчет апостериорного распределения вероятностей для узла исхода при заданных свидетельствах.
   * **Реализация:** Функция predict\_outcome.
     + Проверяются входные аргументы.
     + Фильтруются свидетельства (evidence), оставляя только те, что соответствуют узлам модели.
     + Создается inference = VariableElimination(model).
     + Выполняется result\_distribution = inference.query(variables=[outcome\_node], evidence=valid\_evidence).
     + Результат (распределение) выводится и возвращается вместе со словарем конкретных вероятностей.
3. **Оценка и Ранжирование Влияния Факторов:**
   * **Описание:** Реализация алгоритма анализа чувствительности для ранжирования факторов по их влиянию на неблагоприятный исход в контексте заданных свидетельств.
   * **Реализация:** Функция assess\_factor\_influence.
     + Проверяются входные аргументы.
     + Определяются коды неблагоприятных исходов.
     + Рассчитывается базовая вероятность неблагоприятного исхода P\_{base} с учетом base\_evidence.
     + В цикле для каждого фактора X\_i:
       - Определяется его "проблемное" состояние j.
       - Формируются свидетельства E\_i, где X\_i=j.
       - Рассчитывается условная вероятность P\_i = P(Y>0 | E\_i).
       - Рассчитывается коэффициент влияния Influence\_Ratio\_i = P\_i / P\_{base}.
     + Результаты сохраняются в словарь factor\_influence.
     + Словарь сортируется по убыванию Influence\_Ratio\_i.
     + Возвращается ранжированный список.



1. **Основной Блок (if \_\_name\_\_ == "\_\_main\_\_":)**
   * Последовательно вызывает функции загрузки, предсказания исхода и оценки влияния.
   * Выводит результаты в консоль, включая ранжированную таблицу влияния факторов и наиболее вероятный исход.

Таким образом, программный продукт разделен на модуль обучения и модуль применения, что позволяет гибко использовать обученную модель для анализа различных сценариев без необходимости повторного обучения. Выбор библиотек и методов обоснован их функциональностью и эффективностью для задач вероятностного моделирования и анализа данных.

Программный модуль обеспечивает автоматизацию процесса построения и обучения БСД, включая необходимые проверки консистентности данных и модели.

**3.2.4 Тестирование**

Тестирование программного модуля проводилось на сформированном наборе данных из 146 авиационных событий (aviation\_incidents\_refined.csv).

* Тест 1: Загрузка и валидация. Модуль успешно загрузил данные и файл конфигурации. Первичная валидация выявила отсутствие узла Ext\_UnlawfulInterference в файле bn\_config.json. После добавления данного узла в конфигурацию, валидация прошла успешно, подтвердив соответствие данных и JSON.
* Тест 2: Очистка данных. Данные были успешно преобразованы в целочисленный формат. Пропущенные значения были заменены нулями (с выводом соответствующего предупреждения).
* Тест 3: Создание структуры. Реалистичная структура БСД была успешно создана и прошла проверку на ацикличность.
* Тест 4: Валидация перед обучением. Проверка кардинальностей и диапазонов значений прошла успешно. Были выведены предупреждения для узлов, имеющих только одно состояние в наборе данных (например, HF\_ATC\_CommError, HF\_Ground\_LoadError), что является ограничением данных, а не ошибкой модуля. Финальная проверка на отрицательные значения также прошла успешно.
* Тест 5: Обучение параметров. Метод model.fit() успешно обучил CPT для всех узлов модели без вывода ошибок ValueError или других критических проблем.
* Тест 6: Анализ вероятностей. Модуль корректно рассчитал и вывел априорные и условные вероятности для исхода и выбранных факторов. Результаты анализа показали, например, что отказ техники (Tech\_SystemFailure) и плохие условия на ВПП (Env\_RunwayCondition) значительно повышают вероятность тяжелых исходов (аварии/катастрофы), что соответствует ожиданиям. Некачественная противообледенительная обработка (HF\_Ground\_Deicing), хоть и редко встречалась в данных (2 случая), приводила к тяжелым последствиям (1 авария, 1 катастрофа).
* Тест 7:Логический вывод. Инициализация VariableElimination прошла успешно (ошибка "No CPD associated with Env\_Temperature", возникавшая ранее, была, по-видимому, устранена на этапе подготовки данных или исправления конфигурации). Примеры запросов к модели были выполнены, демонстрируя возможность получения условных вероятностей для заданных сценариев.
* Тест 8: Визуализация. (Выполнено с установленными зависимостями) Модуль успешно сгенерировал и сохранил графическое представление структуры сети в файл realistic\_bn\_structure.png.

Вывод по тестированию: Программный модуль продемонстрировал корректную работу на всех этапах: от загрузки и валидации данных до обучения параметров БСД и выполнения базового анализа и логического вывода. Модель успешно обучается на подготовленных данных и готова к дальнейшему использованию для анализа рисков АП.

**3.3 Анализ результатов**

**ЗАКЛЮЧЕНИЕ**

Целью бакалаврской работы являлась разработка

Для достижения поставленной цели были решены следующие задачи:

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

**Приложение А. Программный код**

Структура проекта: