Интеллектуальная система мультимодального нейросетевого мониторинга активностей

Р. Р. Миннеахметов

***Аннотация*—Предложен подход к созданию интеллектуальной системы мониторинга активности на основе больших языковых моделей. Особое внимание уделяется использованию современных нейросетей и методов компьютерного зрения для комплексного анализа данных видеонаблюдения, сигналов датчиков и журналов событий. В качестве платформы реализации выбран локальный фреймворк Ollama, позволяющий автономно запускать большие языковые модели. Разработан прототип системы; описаны его архитектура, процесс обработки разнородных данных и результаты экспериментальной оценки. Полученные результаты показывают, что применение больших нейросетевых моделей позволяет автоматизировать анализ мультимодальных данных и повышает точность обнаружения аномалий в рассматриваемых сценариях.**

***Ключевые слова*—Ollama, большие нейросетевые модели, мониторинг активности, мультимодальный анализ, видеоаналитика, искусственный интеллект**

I. Введение

За последнее время большие языковые модели (Large Language Models, LLM), продемонстрировали выдающиеся результаты во множестве задач — от понимания естественного языка до анализа данных и поддержки принятия решений [1][2]. Прогресс в области глубокого обучения в сфере компьютерного зрения также привёл к существенным достижениям в распознавании объектов и действий человека на видео [3][4]. Однако многие прикладные задачи мониторинга и обеспечения безопасности требуют одновременной обработки разнотипных данных (визуальных, сенсорных, текстовых), что традиционно выполнялось разрозненными специализированными методами [5][6]. Например, для контроля активности персонала и выявления инцидентов на промышленном объекте необходимо анализировать как видеозаписи с камер наблюдения, так и показания носимых сенсоров, а также данные систем контроля доступа. Существующие подходы, как правило, ограничиваются одним типом источников: сенсорные данные [7][8], видеопоток [9][10][11], журнал событий [23] и т.д. Это создаёт разрыв в целостной оценке обстановки и затрудняет своевременное обнаружение сложных аномалий.

Статья получена 23 октября 2025.

Р. Р. Миннеахметов, Казанский Федеральный Университет, г. Казань, ул. Кремлевская, 35 (e-mail: razil0071999@gmail.com).

Недавно появилось множество работ, предлагающие задействовать возможности больших нейросетевых моделей для отдельных видов данных. Так, в [1] исследуется применение LLM для распознавания активности по данным носимых устройств, а в [10] представлен подход LogGPT, использующий языковую модель для выявления аномалий в лог-файлах. Тем не менее, задача единого анализа мультимодальных данных при помощи нейросетей остаётся недостаточно изученной. Большие языковые модели обладают способностью обобщать знания и гибко интерпретировать текстовые описания сложных ситуаций, а появление мультимодальных версий таких моделей расширило их возможности на визуальную информацию [11][12]. Настоящая работа направлена на сокращение указанного пробела: исследуется, насколько эффективно современные нейросетевые модели могут анализировать и интегрировать разнородные данные (изображения, параметры сенсоров, текстовые события) для обнаружения нештатных ситуаций. В работе представлена архитектура прототипа интеллектуальной системы мониторинга, реализованной на основе LLM, а также приведены результаты экспериментальной оценки её точности и производительности. В последующих разделах изложены постановка задачи, обзор существующих решений, описание предлагаемого метода, ход эксперимента, обсуждение результатов и заключение.

II. Обзор существующих решений

Для эффективного мониторинга активности необходимо учитывать различные типы данных: видеопотоки, системные журналы, сигналы носимых сенсоров, а также контент, создаваемый пользователями (User Generated Content, UGC). Предлагаемый подход базируется на использовании предобученных больших нейросетевых моделей, способных обработать такие разнородные источники и выявлять в них значимые закономерности [9].

A. Анализ видеоданных и сенсорных сигналов

В задачах компьютерного зрения сегодня успешно применяются глубокие сверточные нейросети и Vision Transformer-модели для распознавания действий на видео – они способны классифицировать поведение людей практически в реальном времени [8]. При

обработке последовательностей сенсорных измерений (например, акселерометра и гироскопа) эффективны рекуррентные архитектуры на базе LSTM/GRU или трансформеры, обученные на больших массивах данных о движениях. Эти модели выделяют характерные паттерны, соответствующие различным видам активности (ходьба, бег, падение и др.), и распознают отклонения от нормального поведения [13].

B. Обработка текстовых логов

Для анализа текстовых записей и файлов журналов перспективно применение больших языковых моделей. Рассматривая последовательность событий как текст на естественном языке, LLM способны по контексту обнаруживать аномальные или критические ситуации [3][10].

Рассмотренные нейросетевые подходы уже находят применение во многих областях. Ниже приведены некоторые примеры:

Промышленная безопасность. Нейросетевые модели компьютерного зрения используются для контроля действий работников на производстве и соблюдения техники безопасности. С их помощью в реальном времени выявляется отсутствие каски или другого защитного снаряжения у сотрудника [14], что позволяет мгновенно реагировать и предотвращать инциденты. Анализ вибраций оборудования и других сенсорных данных на основе рекуррентных сетей помогает реализовать предиктивное обслуживание, выявляя отклонения в работе машин и предупреждая аварии [15].

Умные дома и носимые устройства. В бытовой среде интеллектуальные модели мониторят повседневную активность жителей для повышения удобства и безопасности. Например, обработка видеопотока с домашней камеры вместе с сигналами датчиков движения позволяет определить, что пожилой человек упал, и автоматически вызвать помощь. Фитнес-браслеты и смарт-часы с моделями распознавания активности человека (Human Activity Recognition, HAR) отслеживают показатели активности и здоровья пользователя, посылая уведомления при обнаружении отклонений (продолжительная неподвижность, аритмия и т.п.) [16].

Системы наблюдения и кибербезопасность. Методы глубокого обучения внедряются в видеонаблюдение для распознавания подозрительного поведения или опасных ситуаций. Алгоритмы могут обнаружить оставленный без присмотра предмет или агрессивные действия в толпе, предотвращая правонарушения [4]. В сфере информационной безопасности языковые модели анализируют сетевые журналы на наличие подозрительных паттернов, предшествующих атакам, что позволяет оперативно реагировать на киберинциденты [6].

Медицина и здравоохранение. Обработка потоковых данных от носимых сенсоров и анализ речи или текстовых записей пациентов с помощью LLM открывают возможности для раннего выявления признаков ухудшения состояния – таких как стресс, депрессия или начало физических недугов [17][1]. Таким образом, опыт разных индустрий демонстрирует универсальность масштабных нейросетевых моделей: от производственных цехов до домашней обстановки они повышают эффективность мониторинга и снижают влияние человеческого фактора.

III. Постановка задачи

Целью данной работы является разработка прототипа интеллектуальной системы мониторинга активности на основе больших языковых моделей, способной анализировать мультимодальные данные (видеоизображения, показания сенсоров, текстовые логи) для обнаружения нештатных либо аномальных ситуаций. Для достижения этой цели были решены следующие задачи:

1) *спроектирована архитектура системы, объединяющая разнородные источники данных и обеспечивающая их поэтапную обработку с помощью больших моделей;*
2) *адаптированы и применены несколько предобученных крупных моделей (языковых и мультимодальных) для анализа искусственно смоделированных сценариев активности;*
3) *проведена оценка точности распознавания событий и аномалий по каждому сценарию (с использованием метрик, таких как F1-мера) и исследована производительность моделей (время отклика) при локальном развёртывании.*

Основные исследовательские вопросы включают пригодность и точность современных LLM при анализе нетипичных мультимодальных данных, их способность обрабатывать комбинацию визуальных наблюдений и числовых показателей, а также ограничения по быстродействию и ресурсам при практическом применении предлагаемого подхода.

IV. Архитектура системы

Разработанная система мониторинга включает несколько модулей, выполняющих сбор данных, их предварительную обработку и анализ с помощью крупных нейросетевых моделей. Общий процесс функционирования прототипа можно разделить на следующие этапы:

A. Сбор данных

Система собирает синхронизированные данные из трёх источников: видеокамеры (отдельные кадры из короткие видеосегменты), комплекта датчиков температуры и влажности и системы контроля доступа (журнальные записи событий).

B. Предварительная обработка и формирование описаний

На этом этапе исходные данные преобразуются в форму, удобную для восприятия языковой моделью. Визуальные данные напрямую подаются напрямую в модель в виде изображения. Показания сенсоров представляются в структурированном формате

(например, JSON с полями *temperature* и *humidity*), отражающем текущие значения параметров. Записи текстовых логов (например, события системы доступа) при необходимости фильтруются по релевантности и форматируются (время приводится к стандарту ISO 8601 [18], событие описывается в поле *event*). В результате предобработки получается набор текстовых фрагментов, одновременно представляющих содержимое видеокадра, состояние сенсоров и недавние события.

C. Формирование запроса (prompt)

Подготовленные текстовые описания объединяются в единый запрос к языковой модели. Промпт составляется так, чтобы модель получила максимально полный контекст ситуации. Например, запрос может включать словесное описание сцены (или прямую ссылку на изображение, если модель умеет его принимать), текущие показания датчиков с указанием времени, а также последнее событие из лога. Завершается промпт инструкцией, требующей от модели определить, нормальна ли ситуация или имеются признаки аномалии, и обосновать свой вывод. При необходимости модель запрашивается выдавать ответ в формате JSON (через параметр format API Ollama [19]) для облегчения автоматического разбора результата.

D. Анализ модели

Выбранная нейросетевая модель обрабатывает сформированный промпт, опираясь на своё обученное знание, и генерирует ответ. Если используется мультимодальная модель, она анализирует изображение при помощи внутренних визуальных компонентов; иначе модель опирается только на текстовое описание кадра. Ответ может представлять собой либо связное описание ситуации, либо структурированный вывод (например, включает поле *"anomaly": true* с пояснением). В прототипе вызов моделей осуществляется через API фреймворка Ollama [19][20] с использованием официальной Python-библиотеки [21]: запрос содержит идентификатор модели, текст промпта и набор параметров генерации.

E. Интерпретация результата

Сгенерированный моделью ответ анализируется системой. Если получена структурированная строка JSON, из неё извлекаются ключевые поля (например, флаг обнаружения отклонения). Если ответ представлен в свободной текстовой форме, система применяет набор правил или дополнительный запрос к модели для его интерпретации. После этого распознанное событие и его статус (норма или аномалия) фиксируются. Данные этого этапа могут быть использованы для оповещения оператора либо для последующего обучения системы.

Прототип системы развёрнут на локальной платформе **Ollama**, что позволяет загружать и выполнять большие модели без передачи данных внешним сервисам [20]. В качестве ядра анализа протестированы несколько готовых предобученных моделей из каталога Ollama: **gemma3:12b** [22], **LLaVA:13b** [11], **llama3.2-vision:11b** [12] и **minicpm-v:8b** [23]. Выбор пал именно на эти модели, т.к. все они обладают встроенной способностью обрабатывать визуальную информацию благодаря обучению на парах «изображение–текст». Все указанные модели содержат от 8 до 13 миллиардов параметров и предобучены на больших объемах данных. Специального дообучения под поставленную задачу не проводилось — модели применялись как есть, что позволило проверить их базовые возможности в мультимодальном понимании данных.

Важным аспектом настройки системы является выбор параметров генерации ответа модели. API Ollama позволяет регулировать степень случайности ответа через параметр *temperature* и ограничивать выбор генерируемых токенов с помощью *top_p* (nucleus sampling) [19]. Кроме того, можно задать максимальную длину ответа и коэффициент штрафа за повторения (repeat_penalty) для предотвращения избыточного повторения фраз. В проведённых экспериментах, стремясь к воспроизводимости, основное внимание уделялось качеству содержательных ответов, поэтому параметры генерации выбирались консервативно: температура близка к 0 (для получения детерминированного вывода), top_p = 1 (учитывается полный диапазон вероятностей), фиксированный максимальный размер ответа и отключённая потоковая генерация. Такие настройки минимизировали случайные вариации и упростили сравнение моделей по точности и скорости.

V. Экспериментальная часть

A. Генерация тестовых данных

Поскольку не существует готового датасета, содержащего синхронные видеозаписи, сенсоры и логи, все необходимые данные были сгенерированы искусственно. Целью генерации было воспроизвести типичные ситуации, встречающиеся при мониторинге безопасности, включая как нормальные, так и аномальные события. В рамках эксперимента смоделировано шесть сценариев в условиях офисного коридора, контролируемого камерой и датчиками:

1) *Сценарий 1. Падение человека*

Используется изображение с видеокамеры (см. рис. 1) и показания сенсоров (рис. 6). На кадре зафиксирован человек, лежащий без движения на полу. Модель должна распознать признаки аномалии (потеря сознания, травма) и выдать вывод о необходимости немедленной реакции.

2) *Сценарий 2. Возгорание при наличии людей в помещении*

Анализируется видеокадр (рис. 4), сенсорные данные (рис. 7), а также журнальные сообщения системы безопасности, указывающие на пожар. Модель должна учесть повышенные температурные показатели, наличие людей в помещении и лог-сообщения, подтверждающие инцидент, и сделать вывод об аномалии, угрожающей безопасности людей.

3) *Сценарий 3. Возгорание без присутствия людей*

В качестве входных данных используется видеокадр (рис. 5) и параметры сенсоров (рис. 7). Несмотря на то, что сенсоры фиксируют аномальную температуру и снижение влажности, на видео отсутствуют люди. Модель должна распознать факт пожара, но определить, что непосредственной угрозы для человека нет, и тревожная реакция не требуется.

4) Сценарий 4. Попытка проникновения

Используется изображение с камеры (рис. 2), данные сенсоров (рис. 6) и лог, свидетельствующий об открытии входной двери. На видеокадре виден человек, передвигающийся скрытно. В сочетании с сообщением о нарушении доступа, ситуация должна быть классифицирована как аномалия, требующая реакции.

5) Сценарий 5. Акт агрессии

Анализируется кадр видеонаблюдения (рис. 3) и текущие показания сенсоров (рис. 5). На изображении зафиксирован момент нападения одного человека на другого. Модель должна классифицировать ситуацию как критическую и сигнализировать о необходимости вмешательства.

6) Сценарий 6. Штатная ситуация

Используются видеокадр с пустым коридором (рис. 5), нормальные значения сенсоров (рис. 9) и отсутствие логов событий. Ситуация не содержит признаков отклонений, модель должна сделать вывод о нормальном состоянии и не инициировать реагирование.

Для сценариев были сгенерированы статичные изображения, стилизованные под кадры с камер наблюдения (черно-белое изображение с отметкой времени, умеренным шумом и низким разрешением для правдоподобия). Каждый сценарий дополнен синтетическими показаниями датчиков: набор значений, отражающих нормальные условия (например, 22 °C и 45% влажности), и экстремальные при пожаре (например, 85 °C и 10%) . Дополнительно для контекста была сформирована примерная запись системного журнала – обычное событие (вход сотрудника) в формате JSON с полями времени (ISO 8601 [18]) и описанием действия. Таким образом, сформированный тестовый набор охватывает разнообразные источники: визуальный контекст, числовые параметры и текстовые события.

Рисунок 1. Сгенерированное фото с камеры видеонаблюдения. Человек упал.

Рисунок 2. Сгенерированное фото с камеры видеонаблюдения. Человек крадется.

Рисунок 3. Сгенерированное фото с камеры видеонаблюдения. На человека напали.

Рисунок 4. Сгенерированное фото с камеры видеонаблюдения. Человек стоит перед камерой видеонаблюдения.

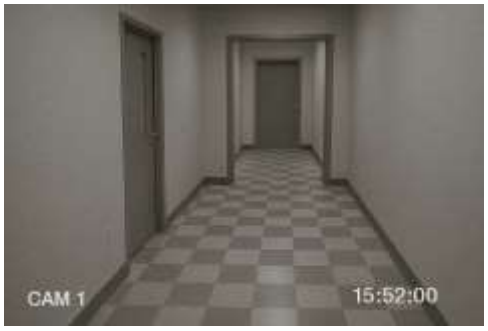

Рисунок 5. Сгенерированное фото с камеры видеонаблюдения. Пустой коридор.


```
{
 "temperature": 26.8,
 "humidity": 33.0
}
```


Рисунок 6. Показания с датчика температуры и влажности. Нормальные показатели температуры и влажности.


```
{
 "temperature": 50.4,
 "humidity": 2.0
}
```


Рисунок 7. Показания с датчика температуры и влажности. Сильное превышение температуры и понижение влажности. Вероятная причина - пожар.

B. Проведение эксперимента

Каждый из подготовленных сценариев последовательно подавался на вход всем выбранным моделям. В промпт включался соответствующий сгенерированный кадр вместе с сопутствующей текстовой информацией (показания датчиков, запись журнала). Структура запроса во всех случаях оставалась единой: вначале приводилась информация о текущей обстановке

(например: «На камере видно: человек лежит на полу; Датчики: temperature=85, humidity=10; Событие: дверь открыта сотрудником № 1234»), затем следовал вопрос модели о том, присутствует ли отклонение от нормы или требуется сигнал тревоги. Такой формат взаимодействия выбран потому, что он максимально использует способность LLM понимать описательный текст и устанавливать связи между разнородными фактами [24]. Отправка запросов и получение ответов у всех моделей были автоматизированы с помощью скрипта на Python через API Ollama [21].

Для оценки качества ответов вручную была подготовлена эталонная разметка каждого сценария: отмечалось, является ситуация нормальной или аномальной, и указывался тип инцидента (если присутствует). Например, для сцены с падением эталон – «аномалия (падение человека)», для пустого коридора – «норма». Выходы моделей приводились к упрощённой форме (фиксировалось, сигнализирует ли модель об аномалии или считает ситуацию штатной). На этой основе для каждой модели вычислялись показатели точности распознавания.

VI. Результаты

Основным критерием качества обнаружения аномалий в эксперименте стала F1-мера, сочетающая полноту и точность выявления событий «аномального» класса [25]. Расчёт F1 проводился с использованием функции из библиотеки Scikit-learn [26]. В таблице 1 приведены итоговые значения F1-score для каждой из протестированных моделей (сводно по всем основным сценариям № 1–6).

Таблица 1 – F1-мера обнаружения аномалий для различных моделей

Модель	F1-Score
llama3.2-vision:11b	0.5714285714285714
gemma3:12b	0.8888888888888888
llava:13b	0.0
minicpm-v:8b	0.8

Как видно, качество обнаружения аномалий существенно различается между моделями. Лучший результат показала модель gemma3:12b (F1 ≈ 0,89), правильно идентифицировавшая почти все нештатные ситуации. Немного уступила ей облегчённая minicpm-v:8b (F1 = 0,80), допустившая лишь несколько ошибок. Модель llama3.2-vision:11b верно определила около половины аномалий (F1 ≈ 0,57), тогда как мультимодальная LLaVA:13b фактически не справилась с задачей (F1 ≈ 0), не обнаружив ни одного отклонения. Вероятно, такие контрастные результаты обусловлены отличиями в обучении и настройке моделей. Например, gemma3, несмотря на отсутствие прямого «зрения», сумела корректно интерпретировать текстовые описания сцен, тогда как LLaVA, обладая способностью анализировать изображения, могла неверно понять контекст сценариев или не была обучена распознавать подобные ситуации. Возможно, LLaVA не встречала в обучающей выборке специфических сцен видеонаблюдения, поэтому не уловила даже очевидного признака инцидента (человек, лежащий на полу), который для человека однозначно свидетельствует о проблеме.

Помимо точности классификации, измерялось время отклика каждой модели. В таблице 2 представлены задержки генерации ответа (в секундах) для каждого из шести базовых сценариев, а также суммарное время обработки всего набора.

Таблица 2 – Время генерации ответа (с) по каждому сценарию и в целом

Модель /Сценарий	1	2	3	4	5	6	Общее время
llama3.2-vision:11b	3,36 с	3,12 с	3,06 с	2,57 с	22,82 с	2,89 с	37,83 с
gemma3:12b	14,65 с	11,15 с	10,73 с	10,26 с	10,06 с	9,86 с	66,70 с
llava:13b	12,83 с	7,46 с	4,64 с	5,06 с	5,82 с	4,60 с	40,40 с
minicpm-v:8b	12,62 с	10,59 с	10,95 с	10,31 с	9,83 с	1,88 с	56,18 с

Общее время обработки всех сценариев колеблется от ~38 с (у llama3.2-vision) до ~67 с (у gemma3). Это отражает как различия в объёме моделей (большее число параметров требует больше вычислений), так и особенности их реализации. Модель gemma3:12b оказалась самой медленной, хотя и наиболее точной: её суммарное время ~66,7 с почти вдвое больше, чем у схожей по размеру LLaVA. Вероятно, gemma3 менее оптимизирована по скорости либо генерирует более развёрнутые ответы. Напротив, llama3.2-vision:11b продемонстрировала лучшую скорость, обработав все кейсы примерно за 38 с. Однако у неё наблюдалась наибольшая вариативность времени между разными примерами: если большинство изображений обрабатывались за 2–3 с, то на сценарий 5 (пустой коридор) модель затратила почти 23 с. Очевидно, на пустой сцене LLM сгенерировала более длинное рассуждение, пытаясь осмыслить ситуацию без явных объектов, что увеличило длительность вывода. У модели minicpm-v:8b, наоборот, самый быстрый отклик пришёлся на сценарий 6 (~1,9 с), поскольку в этом случае не требовалось анализировать изображение, и небольшую текстовую задачу компактная модель решила очень быстро. В остальных же случаях minicpm-v уступала по скорости более крупным моделям, возможно из-за менее эффективной архитектуры или отсутствия специализированных оптимизаций. Таким образом, разные модели продемонстрировали различное поведение по быстродействию: более «тяжёлые» способны работать достаточно быстро благодаря оптимизациям, а меньшие по размеру не гарантируют выигрыша, особенно если задача выходит за пределы их базовой специализации.

В целом, результаты подтверждают перспективность

использования больших нейросетевых моделей для анализа мультимодальных данных, но одновременно показывают, что выбор конкретной модели критически влияет на качество и быстродействие системы. Без дополнительной адаптации предобученные модели дают разнородные результаты: одни (например, gemma3) практически из коробки успешно выявляют нештатные ситуации, тогда как другие (например, LLaVA) требуют дообучения или более тщательного подхода к формированию промпта, чтобы обеспечивать полезные выводы. Диапазон значений F1 от 0 до ~0,9 указывает на необходимость тщательного подбора и настройки модели под конкретную предметную область. Аналогично, разброс временных затрат от единиц до десятков секунд означает, что при внедрении такой системы на практике придётся учитывать ограничения по быстродействию и находить компромисс между скоростью и точностью анализа.

VII. Заключение

Созданный прототип интеллектуальной системы мониторинга продемонстрировал принципиальную возможность применения крупных нейросетевых моделей для одновременного анализа разнородных данных (видео, сенсоры, текстовые логи) в задаче отслеживания активности. Показано, что даже без специального обучения на целевой выборке некоторые предобученные модели способны выявлять аномалии, синтезируя информацию из описаний разных модальностей. Это подтверждает актуальность выбранного подхода: современные LLM могут выступать своеобразным «мозгом» системы наблюдения, объединяющим разнообразные сигналы и помогая оператору быстрее получать инсайт о происходящем. Практическая ценность такого решения заключается в снижении необходимости непрерывного ручного контроля и разработки множества узкоспециализированных детекторов – одна универсальная модель либо небольшой ансамбль моделей способен выполнять сразу несколько задач анализа. Помимо этого, использование мультимодальных моделей позволяет учитывать контекст: например, сопоставлять событие из журнала с видеокадром и показаниями датчиков, что повышает надёжность обнаружения комплексных инцидентов.

Научная новизна работы состоит в объединении разрозненных направлений (видеоаналитика, анализ сенсорных потоков, обработка текстовых логов) посредством единого подхода на базе больших языковых моделей. Проведённое исследование выявило как преимущества, так и текущие ограничения такого подхода. К преимуществам относится универсальность и адаптивность крупных моделей – они способны интерпретировать нестандартные описания и делать выводы, близкие к выводам человека. Однако обнаружены и проблемы: разные модели продемонстрировали очень неоднородную эффективность, что указывает на необходимость их адаптации или целевого дообучения под конкретную предметную область. В дальнейшем планируется расширить набор рассматриваемых сценариев и типов данных, а также выполнить обучение моделей на специализированных мультимодальных датасетах, чтобы повысить точность распознавания специфических ситуаций (например, падений, конфликтов и т.д.). Интерес представляет и оптимизация производительности системы: снижение времени отклика (например, за счёт параллельной обработки) и применение иерархических схем с моделями разного масштаба, когда первичную фильтрацию событий выполняет компактная модель, а детальный анализ – более мощная.

В заключение, интеграция больших нейросетевых моделей в системы анализа активности является перспективным направлением, способным повысить интеллектуальный и автономный потенциал средств мониторинга во многих сферах, включая промышленную безопасность, умный дом и кибербезопасность.

Весь исходный код скриптов, сгенерированные тестовые данные и результаты работы моделей доступны в открытом доступе на GitHub: https://github.com/minneakhmetov/llm-activity.

Библиография

[1] E. Ferrara, “Large Language Models for Wearable Sensor-Based Human Activity Recognition, Health Monitoring, and Behavioral Modeling,” *Sensors*, vol. 24, no. 15, p. 5045, 2024.

[2] OpenAI, “*ChatGPT-4o-mini*,” [Online]. Available: https://chatgpt.com/. Accessed: Mar. 30, 2025.

[3] А. В. Пятаева, М. А. Мерко, В. А. Жуковская, и А. А. Казакевич, “Распознавание активности человека по видеоданным,” *International Journal of Advanced Studies*, т. 12, № 4, с. 96–110, 2022.

[4] R. Sharma and N. Patel, “Deep learning-based anomaly detection in surveillance videos,” *J. Vis. Commun. Image Represent.*, vol. 86, p. 103624, 2022.

[5] И. В. Котенко, О. В. Полубелова, И. Б. Саенко, и А. А. Чечулин, “Применение онтологий и логического вывода для управления информацией и событиями безопасности,” *Системы высокой доступности*, т. 8, № 2, с. 100–108, 2012.

[6] B. Nour, M. Pourzandi, and M. Debbabi, “A Survey on Threat Hunting in Enterprise Networks,” *IEEE Commun. Surveys Tuts.*, vol. 25, pp. 2299–2324, 2023. doi: 10.1109/COMST.2023.3299519.

[7] S. Suh, V. F. Rey, and P. Lukowicz, “Tasked: Transformer-based adversarial learning for human activity recognition using wearable sensors,” *Knowl.-Based Syst.*, vol. 260, p. 110143, 2023.

[8] S. Gupta, “Deep learning-based human activity recognition using wearable sensor data,” *Int. J. Inf. Manag. Data Insights*, vol. 1, p. 100046, 2021.

[9] N. D. Nath, A. H. Behzadan, and S. G. Paal, “Deep learning for site safety: Real-time detection of personal protective equipment,” *Autom. Constr.*, vol. 112, p. 103085, 2020.

[10] S. Han, S. Yuan, and M. Trabelsi, “LogGPT: Log Anomaly Detection via GPT,” *arXiv preprint*, 2023. [Online]. Available: https://arxiv.org/pdf/2309.14482.

[11] Ollama, “*llava:13b Model*,” [Online]. Available: https://ollama.com/library/llava:13b. Accessed: Mar. 30, 2025.

[12] Ollama, “*llama3.2-vision:11b Model*,” [Online]. Available: https://ollama.com/library/llama3.2-vision. Accessed: Mar. 30, 2025.

[13] A. Uçar, M. Karakoşe, and N. Kırımça, “Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends,” *Appl. Sci.*, vol. 14, no. 2, p. 898, 2024.

[14] S. Özüağ and Ö. Ertuğrul, “Enhanced Occupational Safety in Agricultural Machinery Factories: Artificial Intelligence-Driven Helmet Detection Using Transfer Learning and Majority Voting,” *Appl. Sci.*, vol. 14, p. 11278, 2024. doi: 10.3390/app142311278.

[15] X. Li, Y. Chen, and L. Hu, “Real-time workplace activity recognition using deep learning models,” *IEEE Trans. Ind. Inf.*, vol. 19, no. 2, pp. 1520–1532, 2023.

[16] Z. Wu, J. Zhao, and H. Shen, “Smart home automation based on human activity recognition: A survey,” *Future Gener. Comput. Syst.*, vol. 137, pp. 41–57, 2023.

[17] S. Yadav, C. K. Jha, and N. Kumar, "AI-powered fall detection systems for elderly care: Challenges and future directions," *Comput. Methods Programs Biomed.*, vol. 230, p. 107416, 2024.
[18] ISO, "*ISO 8601-1:2019 Standard*," [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:8601:-1:ed-1:v1:en. Accessed: Mar. 30, 2025.
[19] Ollama, "*API Documentation*," [Online]. Available: https://github.com/ollama/ollama/blob/main/docs/api.md. Accessed: Mar. 30, 2025.
[20] *Ollama*, [Online]. Available: https://ollama.com/. Accessed: Mar. 30, 2025.
[21] Ollama, "*Python Library*," [Online]. Available: https://github.com/ollama/ollama-python. Accessed: Mar. 30, 2025.
[22] Ollama, "*gemma3:12b Model*," [Online]. Available: https://ollama.com/library/gemma3:12b. Accessed: Mar. 30, 2025.
[23] Ollama, "*minicpm-v:8b Model*," [Online]. Available: https://ollama.com/library/minicpm-v. Accessed: Mar. 30, 2025.
[24] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," *arXiv preprint*, 2024. [Online]. Available: https://arxiv.org/pdf/2402.07927.
[25] D. J. Hand and P. Christen, "F*: an interpretable transformation of the F-measure," *J. Classification*, vol. 38, no. 1, pp. 3–17, 2021.
[26] Scikit-learn, "*F1-Score*," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html. Accessed: Mar. 30, 2025.

Intelligent multimodal neural network activity monitoring system

R. Minneakhmetov

***Abstract*— An approach to creating an intelligent activity monitoring system based on large language models is proposed. Special attention is paid to the use of modern neural networks and computer vision methods for complex analysis of video surveillance data, sensor signals and event logs. The local Ollama framework has been chosen as the implementation platform, which allows large language models to be run independently. A prototype of the system has been developed; its architecture, the process of processing heterogeneous data, and the results of an experimental evaluation are described. The results show that the use of multiple neural network models makes it possible to automate the analysis of multimodal data and increases the accuracy of anomaly detection in the scenarios under consideration.**

***Keywords*—Ollama, large neural network models, activity monitoring, multimodal analysis, video analytics, artificial intelligence**

REFERENCES

[1] E. Ferrara, "Large Language Models for Wearable Sensor-Based Human Activity Recognition, Health Monitoring, and Behavioral Modeling," *Sensors*, vol. 24, no. 15, p. 5045, 2024.

[2] OpenAI, "*ChatGPT-4o-mini*," [Online]. Available: https://chatgpt.com/. Accessed: Mar. 30, 2025.

[3] A. V. Pyataeva, M. A. Merko, V. A. Zhukovskaya, and A. A. Kazakevich, "Recognition of human activity from video data," *International Journal of Advanced Studies*, vol. 12, No. 4, pp. 96-110, 2022.

[4] R. Sharma and N. Patel, "Deep learning-based anomaly detection in surveillance videos," *J. Vis. Commun. Image Represent.*, vol. 86, p. 103624, 2022.

[5] I. V. Kotenko, O. V. Polubelova, I. B. Sayenko, and A. A. Chechulin, "Application of ontologies and logical inference for managing information and security events," *High Availability Systems*, vol. 8, No. 2, pp. 100-108, 2012.

[6] B. Nour, M. Pourzandi, and M. Debbabi, "A Survey on Threat Hunting in Enterprise Networks," *IEEE Commun. Surveys Tuts.*, vol. 25, pp. 2299–2324, 2023. doi: 10.1109/COMST.2023.3299519.

[7] S. Suh, V. F. Rey, and P. Lukowicz, "Tasked: Transformer-based adversarial learning for human activity recognition using wearable sensors," *Knowl.-Based Syst.*, vol. 260, p. 110143, 2023.

[8] S. Gupta, "Deep learning-based human activity recognition using wearable sensor data," *Int. J. Inf. Manag. Data Insights*, vol. 1, p. 100046, 2021.

[9] N. D. Nath, A. H. Behzadan, and S. G. Paal, "Deep learning for site safety: Real-time detection of personal protective equipment," *Autom. Constr.*, vol. 112, p. 103085, 2020.

[10] S. Han, S. Yuan, and M. Trabelsi, "LogGPT: Log Anomaly Detection via GPT," *arXiv preprint*, 2023. [Online]. Available: https://arxiv.org/pdf/2309.14482.

[11] Ollama, "*llava:13b Model*," [Online]. Available: https://ollama.com/library/llava:13b. Accessed: Mar. 30, 2025.

[12] Ollama, "*llama3.2-vision:11b Model*," [Online]. Available: https://ollama.com/library/llama3.2-vision. Accessed: Mar. 30, 2025.

[13] A. Uçar, M. Karakoşe, and N. Kırımça, "Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends," *Appl. Sci.*, vol. 14, no. 2, p. 898, 2024.

[14] S. Özüağ and Ö. Ertuğrul, "Enhanced Occupational Safety in Agricultural Machinery Factories: Artificial Intelligence-Driven Helmet Detection Using Transfer Learning and Majority Voting," *Appl. Sci.*, vol. 14, p. 11278, 2024. doi: 10.3390/app142311278.

[15] X. Li, Y. Chen, and L. Hu, "Real-time workplace activity recognition using deep learning models," *IEEE Trans. Ind. Inf.*, vol. 19, no. 2, pp. 1520–1532, 2023.

[16] Z. Wu, J. Zhao, and H. Shen, "Smart home automation based on human activity recognition: A survey," *Future Gener. Comput. Syst.*, vol. 137, pp. 41–57, 2023.

[17] S. Yadav, C. K. Jha, and N. Kumar, "AI-powered fall detection systems for elderly care: Challenges and future directions," *Comput. Methods Programs Biomed.*, vol. 230, p. 107416, 2024.

[18] ISO, "*ISO 8601-1:2019 Standard*," [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:8601:-1:ed-1:v1:en. Accessed: Mar. 30, 2025.

[19] Ollama, "*API Documentation*," [Online]. Available: https://github.com/ollama/ollama/blob/main/docs/api.md. Accessed: Mar. 30, 2025.

[20] *Ollama*, [Online]. Available: https://ollama.com/. Accessed: Mar. 30, 2025.

[21] Ollama, "*Python Library*," [Online]. Available: https://github.com/ollama/ollama-python. Accessed: Mar. 30, 2025.

[22] Ollama, "*gemma3:12b Model*," [Online]. Available: https://ollama.com/library/gemma3:12b. Accessed: Mar. 30, 2025.

[23] Ollama, "*minicpm-v:8b Model*," [Online]. Available: https://ollama.com/library/minicpm-v. Accessed: Mar. 30, 2025.

[24] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," *arXiv preprint*, 2024. [Online]. Available: https://arxiv.org/pdf/2402.07927.

[25] D. J. Hand and P. Christen, "F*: an interpretable transformation of the F-measure," *J. Classification*, vol. 38, no. 1, pp. 3–17, 2021.

[26] Scikit-learn, "*F1-Score*," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html. Accessed: Mar. 30, 2025