

А. АЛОС, З. ДАХРУДЖ

## ОБНАРУЖЕНИЕ КОНТЕКСТНЫХ НЕИСПРАВНОСТЕЙ В БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТАХ С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКОЙ ЛИНЕЙНОЙ РЕГРЕССИИ И КЛАССИФИКАЦИИ МЕТОДОМ К-БЛИЖАЙШИХ СОСЕДЕЙ

*Беспилотный летательный аппарат (БПЛА) представляет собой сложную систему, при проектировании которой рассматриваются проблемы управления, аэродинамики и связи. В статье предложен новый метод обнаружения контекстных неисправностей на основе сложных линейных соотношений между параметрами БПЛА (показаниями датчиков и командами). Под контекстными неисправностями подразумеваются вырабатываемые неисправным датчиком значения, недопустимые в контексте других параметров. Предлагаемый подход основан на оценке значений целевого параметра с использованием динамической линейной регрессии, после чего выполняется расчет погрешности оценивания на каждом временном интервале. Путем классификации методом ближайших соседей (K-Nearest Neighbour – K-NN) значения погрешности оценивания разделяются на нормальные и аномальные. Аномальные значения принимаются за потенциально ошибочные. Помимо этого, предложенный метод сравнивается с другими методами поиска аномалий – k-средних (K-Means) и One-Class SVM. Результаты сопоставления продемонстрировали более высокую эффективность нового подхода в большинстве случаев.*

**Ключевые слова:** БПЛА, полет, линейная регрессия, обнаружение аномалий, аномальный, K-NN, классификация.

### Введение

БПЛА представляет собой управляемый дистанционно или с помощью бортовых компьютеров летательный аппарат [1], который нашел широкое применение в различных областях. Сложность конструкции БПЛА увеличивает вероятность его выхода из строя. Во время полета сохраняются значения его параметров для фиксированных интервалов времени. Под параметрами понимаются команды и показания датчиков. Команды – это непрерывные управляющие сигналы, с помощью которых задаются, в частности, высота, направление, крен и др. К показаниям датчиков относятся данные о скорости полета, углах тангажа, крена, рыскания и т.п. Алгоритмы обнаружения аномалий позволяют прогнозировать отказ системы за счет выявления комбинаций данных, не соответствующих ожидаемым значениям [2]. Такие алгоритмы работают в одном из трех режимов: контролируемом, полуконтролируемом

**Алос** Ахмад. Аспирант, кафедра информатики, Высший институт прикладных наук и технологий (Дамаск, Сирия).

**Дахрудж** Зухайр. Доктор наук, преподаватель, кафедра информатики, Высший институт прикладных наук и технологий.

Научный редактор перевода д.т.н. Н.В. Колесов.

и неконтролируемом [2, 3]. При использовании алгоритмов в контролируемом режиме предполагается наличие обучающих данных с заданными метками как для нормального, так и для аномального класса. В случае применения алгоритмов в полуконтролируемом режиме необходимо наличие обучающих данных с метками, относящимися только к нормальному классу. Для алгоритмов обнаружения аномалий в неконтролируемом режиме обучающие данные с метками не требуются. При этом подразумевается, что количество нормальных событий значительно превосходит число аномалий. В общем случае неисправности датчиков БПЛА можно разделить на три типа: точечные, контекстные и совокупные [2]. Точечный сбой означает, что датчик вырабатывает неверное или недопустимое значение. Контекстная неисправность возникает, когда датчик вырабатывает значение, недопустимое относительно текущего контекста, например, нулевую высоту при нахождении БПЛА в воздухе. При совокупной неисправности датчик вырабатывает целый ряд неверных значений. Например, БПЛА снижается, но высотомер показывает фиксированное значение [4]. Представим сценарий, когда пилот принимает решение увеличить высоту, чтобы предотвратить падение БПЛА, летящего критически низко. Если команда оказалась крайне неэффективной, БПЛА, разумеется, разобьется. В этом сценарии важны два параметра: команда управления высотой и показания датчика высоты. Предложенный метод позволяет воспользоваться локальным линейным соотношением между каждой парой непрерывных параметров. Назовем один из них зависимым, а второй – целевым. Целевой параметр связан с потенциально неисправным датчиком, который мы пытаемся найти, зависимый – с контекстом неисправности. Представленный метод предусматривает поиск аномалий в значениях целевого параметра. Обнаруженные аномалии могут привести к потенциальной неисправности соответствующего датчика. В описанном сценарии команда управления высотой является зависимым параметром (который считается непрерывным), а показания датчика высоты, в котором проявилась неисправность, – целевым. Алгоритм предполагает, что значения всех параметров (показаний датчиков и команд) являются непрерывными вещественными числами, а не кодами или импульсами. Используемый метод скользящего окна [5] задействует ограниченное число предшествующих точек. Эти точки служат для оценки значений целевого параметра с помощью динамической линейной регрессии (*dynamic linear regression – DLR*), которая применяется к точкам скользящего окна. Всякий раз при перемещении скользящего окна линейные соотношения изменяются. Значения погрешности оценивания классифицируются с помощью классификатора *K-NN*. Согласно этой классификации, соответствующие точки скользящего окна помечаются как нормальные или аномальные [6]. Мы протестировали алгоритм на модельных полетных данных и сравнили наш метод с другими алгоритмами, в частности *k-средних* и *One-Class SVM*. В большинстве случаев наш алгоритм продемонстрировал лучшие результаты.

## **Обзор литературы**

Существует три типа алгоритмов обнаружения неисправностей: один основан на использовании модели, второй – знаний, третий – на управлении данными. Так, в работе [8] описывается алгоритм обнаружения неисправности первого типа, который позволял выполнить оценку состояния БПЛА с помощью нелинейной динамической

модели летательного аппарата. Для обнаружения погрешности авторы использовали расхождение оценки состояния и его текущего значения. Алгоритмы, основанные на знаниях, записываются на языке продукционных правил («если..., то...»). Авторы [9] разработали метод обнаружения неисправностей датчиков БПЛА с помощью модели с нечеткой логикой. И, наконец, в основе третьего типа алгоритмов лежит статистическая информация, позволяющая выявить выбросы (outliers) и обозначить их как сбои. В [10] был разработан онлайн-алгоритм для выявления неисправностей датчиков БПЛА на основе статистического анализа их показаний и навигационных данных. Авторы [4] акцентировали внимание на алгоритмах на основе управления данными, поскольку другие типы алгоритмов оказались неспособны определить неизвестные или несмоделированные сбои. Алгоритмы на основе использования моделей и знаний сильно зависят от моделей и правил, тогда как алгоритмы на основе управления данными продемонстрировали высокое быстродействие и адаптивность благодаря безмодельному анализу. В работах [10–12] применялись статистические методы, с помощью которых были получены оценки аномалий для каждой заданной точки. Оценка аномалии зависит от набора отслеженных данных и размера окна. Такие методы учитывают плотность точек, например расстояние Махаланобиса или K-NN. Во многих алгоритмах обнаружения аномалий широко применялся анализ главных компонент (АГК), который представляет собой быстрый способ снижения размерности. Он позволяет уменьшить размерность многомерных наборов данных до двух-трех параметров. Авторы [13] разработали детектор аномалий на основе АГК для прогнозирования сетевых кибератак. В [14] рассмотрено применение АГК для обнаружения аномалий при большом объеме выборки и сложной корреляции данных датчиков БПЛА. Большинство методов обнаружения аномалий имеют заранее определенные ограничения, но в реальных системах данным часто свойственны неоднородность, многомерность и наличие шумов. В работе [15] описывается метод задания порога динамической погрешности, который позволяет классифицировать аномальные значения. При этом для преодоления разнообразия применялся неконтролируемый алгоритм, а в потоки данных были включены параметры шумов. В следующем разделе дано описание предлагаемого алгоритма, а также используемых в настоящей работе методов.

## Методы

*Постановка задачи.* Во время каждого полета БПЛА передает значения своих параметров на наземную станцию. Система собирает данные и хранит их в последовательных строках на каждом временном интервале  $t$ . Пусть  $P$  – это множество всех целевых параметров, а  $Q$  – множество всех зависимых параметров. Примем  $p_j \in P, q_k \in Q$  как параметры с индексами  $j, k$  для соответствующих множеств. Записанными значениями для  $(q_k, p_j)$  на временном интервале  $t$  являются  $(x_{(q_k)}^t, x_{(p_j)}^t)$ . Предположим, что мы использовали скользящее окно с размером  $h$ . Скользящее окно состоит из предшествующих точек пары  $(q_k, p_j)$  между шагом  $t-h$  и  $t$ . Точки скользящего окна сохраняются в динамическом векторе

$$W_{q_k, p_j}^{t, h} = \left[ \left( x_{q_k}^{t-h}, x_{p_j}^{t-h} \right), \left( x_{q_k}^{t-h+1}, x_{p_j}^{t-h+1} \right), \dots, \left( x_{q_k}^t, x_{p_j}^t \right) \right].$$

Задача – поиск аномалий в значениях каждого параметра  $p_j$  в контексте параметра  $q_k$ , как показано на рис. 1.

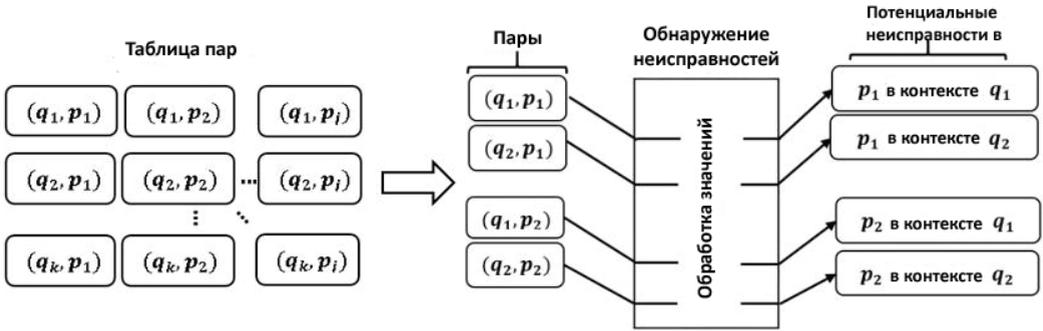


Рис. 1. Обнаружение контекстных неисправностей с использованием пар параметров БПЛА

Алгоритм DLR–KNN начинается с оценивания значений  $p_j$  с помощью динамической линейной регрессии. При этом используются точки скользящего окна  $W_{q_k, p_j}^{t, h}$ . Алгоритм вычисляет значения погрешности оценивания. После этого алгоритм K-NN пытается извлечь аномальные значения погрешности оценивания, которые будут использованы для маркировки соответствующих точек в качестве потенциальных неисправностей (см. рис. 2). Для удобства чтения формул далее по тексту вместо  $p_j$  и  $q_k$  мы будем использовать соответственно  $p$  и  $q$ .

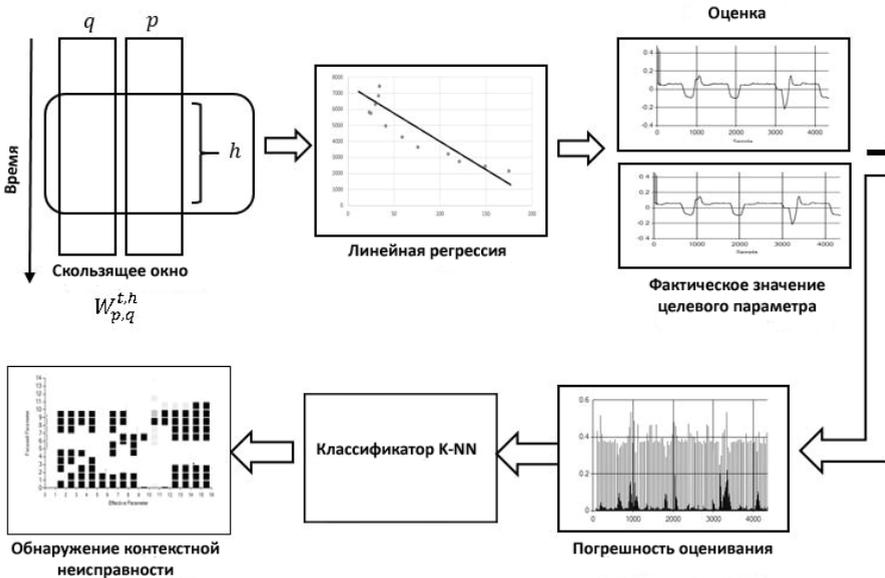


Рис. 2. Общая схема метода обнаружения аномалий DLR–KNN

**Динамическая линейная регрессия и локальная корреляция.** Статистический метод линейной регрессии позволяет охарактеризовать отношение между двумя непрерывными переменными, определив наиболее подходящую линию, проходящую через точки двух переменных [16]. Вычислим наклон  $b'$  и координату  $a'$  пересечения

оси у линии динамической линейной регрессии для точек  $W_{q_k, p_j}^{t, h}$  по формулам (1) и (2), где  $\bar{x}_p$  – среднее значение  $x_p^i$ ,  $\bar{x}_q$  – среднее значение  $x_q^i$  между шагом  $t-h$  и  $t$ .

$$b^t = \frac{\sum_{i=t-h}^t (x_q^i - \bar{x}_q)(x_p^i - \bar{x}_p)}{\sqrt{\sum_{i=t-h}^t (x_q^i - \bar{x}_q)^2}}, \quad (1)$$

$$a^t = \bar{x}_p - \frac{\sum_{i=t-h}^t (x_q^i - \bar{x}_q)(x_p^i - \bar{x}_p)}{\sqrt{\sum_{i=t-h}^t (x_q^i - \bar{x}_q)^2}} \bar{x}_q \quad (2)$$

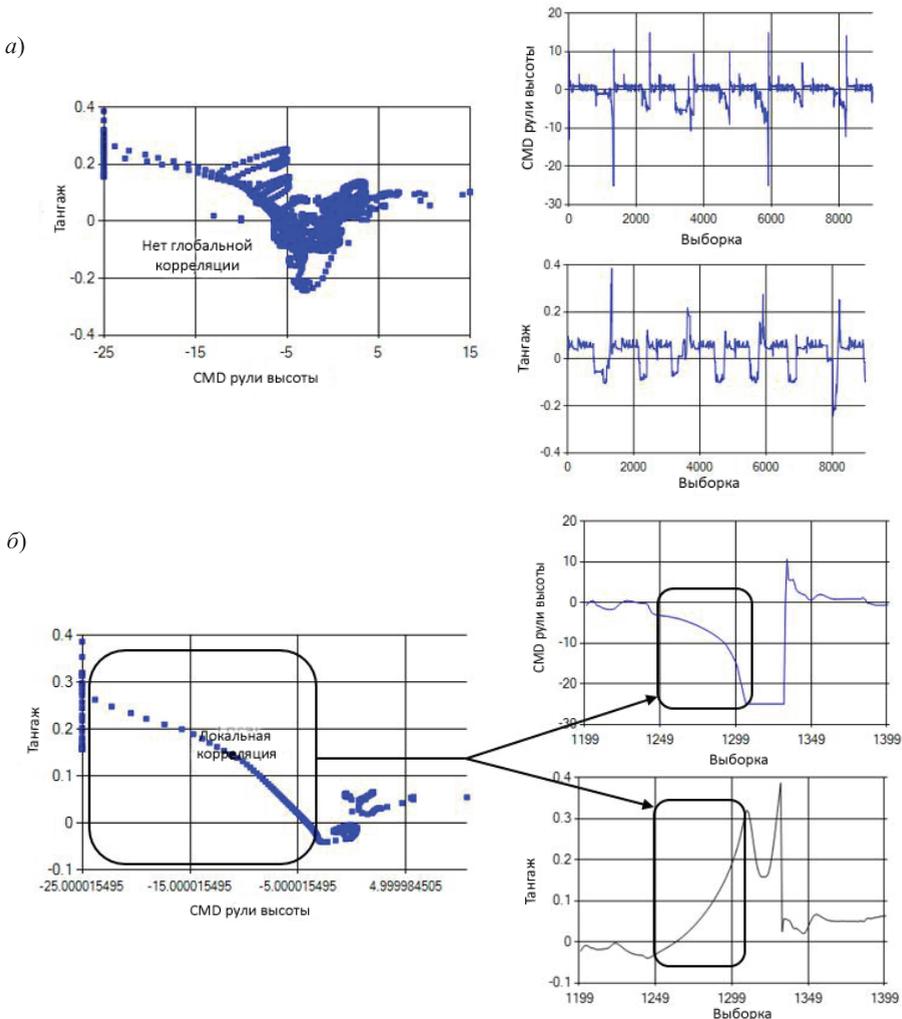


Рис. 3. Сравнение локальной и глобальной корреляции параметров:  
 а – угол тангажа и команда управления высотой глобально не коррелируют;  
 б – имеется локальная корреляция между углом тангажа и командой управления высотой при использовании скользящего окна

В таких сложных системах, как БПЛА, многие параметры из различных источников коррелируют друг с другом весьма замысловатым образом. При этом во время полета корреляция меняется, в результате соотношения и линейная регрессия тоже будут меняться. Например, при полете в неподвижном и устойчивом воздухе фактическая высота будет коррелировать с командой управления высотой. При входе в зону турбулентности с относительно большой вертикальной составляющей БПЛА будет подниматься с восходящими потоками воздуха и опускаться с нисходящими. В каждом состоянии автопилот должен реагировать на эти изменения, и корреляция между заданными командами высотой и фактическим ее значением начнет изменяться. Кроме того, во многих случаях будет наблюдаться отставание по фазе и запаздывание между входными и выходными данными. В работе [11] это поясняется на небольшом примере. Предлагаемый нами метод позволяет решить проблему запаздываний и варьирования корреляции: на каждом временном интервале алгоритм использует скользящее окно, которое состоит из небольшого количества точек полетных данных. На протяжении этого окна локальная корреляция и линейное соотношение считаются устойчивыми, а запаздывание между входными и выходными данными – фиксированным, за счет этого влиянием запаздывания можно пренебречь. На рис. 3, а показано, что значения команды управления высотой и угол тангажа глобально не коррелируют, поэтому линейная регрессия здесь не слишком эффективна. Вместе с тем на рис. 3, б два предыдущих параметра сильно коррелируют на интервале времени между 1250 и 1300 с. Эта корреляция устанавливается эмпирически с помощью локальной линейной зависимости, где может применяться динамическая линейная регрессия.

**Обнаружение аномалий.** Алгоритм DLR–KNN находит аномальные значения  $p$  в контексте  $q$  (см. алгоритм 1). Алгоритм строит скользящее окно, вычисляет наклон и координату пересечения линейной регрессии и оценивает значение  $\hat{x}_p^t$  для целевого параметра  $p$  на временном интервале  $t$  по формуле (3). Затем он вычисляет погрешность оценивания  $e_p^t$  по формуле (4):

$$\hat{x}_p^t = b^t x_q^t + a^t, \quad (3)$$

$$e_p^t = |x_p^t - \hat{x}_p^t|. \quad (4)$$

Метод K-NN используется в алгоритме для классификации значений погрешности оценивания. Как правило, K-NN применяется для решения задач классификации путем оценивания локальной плотности точек данных [17], которая зависит от расстояния между тестируемым значением и его «ближайшими соседями». Предположим,  $NN_k(e_p^t)$  является набором данных о  $k$ -ближайших соседях для текущей погрешности прогнозирования, а  $NN_k>NN_k(e_p^t)$  –  $k$ -ближайшими соседями  $NN_k(e_p^t)$ . Тогда алгоритм K-NN использует неравенство

$$\left( \frac{\|e_p^t - NN_k(e_p^t)\|}{\|NN_k(e_p^t) - NN_k>NN_k(e_p^t)\|} \leq 1 \right)$$

для определения класса  $e_p^t$ . Если неравенство верно, то точка относится к тому же классу, что и  $NN_k(e_p^t)$  [5].

**Алгоритм 1 для определения аномалий  $p, q$  (методом DLR–KNN)** $E_p \leftarrow \emptyset$ **Пока**  $t <$  *общего времени набора данных*, **выполнить**:*Извлечь скользящее окно  $W_{q,p}^{t,h}$ , в котором все точки нормальные**Вычислить  $b^t$  и  $a^t$* 

$$\hat{x}_p^t \leftarrow b^t x_q^t + a^t$$

$$e_p^t \leftarrow |x_p^t - \hat{x}_p^t|$$

*Сгладить  $e_p^t$  с помощью  $E_p$* *Классифицировать  $e_p^t$  по алгоритму K-NN***Если**  $e_p^t \in$  *класс аномалий*, **то***пометить  $x_p^t$  как аномальное значение  $p$  в контексте  $q$* **Если нет**,*прибавить  $e_p^t$  к  $E_p$* **Вернуться** (*аномальные значения*)

В нашей реализации используется евклидово расстояние  $\|\cdot\|$ , но можно применять и другие функции расстояния, например манхэттенское и Хэмминга [17]. Если точка нормальна, алгоритм DLR–KNN добавляет  $e_p^t$  к вектору погрешности  $E_p = [e_p^0, e_p^1, \dots, e_p^{(\text{общее время})}]$ . Вектор погрешности используется при сглаживании значений погрешности, которое позволяет уменьшить большой разброс значений в случае резких изменений. Простым способом сглаживания значений  $e_p^t$  является усреднение вышеуказанных  $k$  значений погрешности, где точки уже отнесены к классу нормальных

$$\left( \frac{1}{k} \sum_{j=t-k}^{j=t} e_p^j \right).$$

В то же время скользящее окно может содержать некоторые из ранее обнаруженных аномальных точек, которые влияют на оценивание с помощью линейной регрессии, поскольку увеличивают значение погрешности (рис. 4). Соответственно, нормальные точки могут быть отнесены к аномальным. Для решения этой проблемы нужно пренебречь выявленными аномальными точками из построенного скользящего окна, тогда окно будет состоять из последних  $h$  нормальных точек.

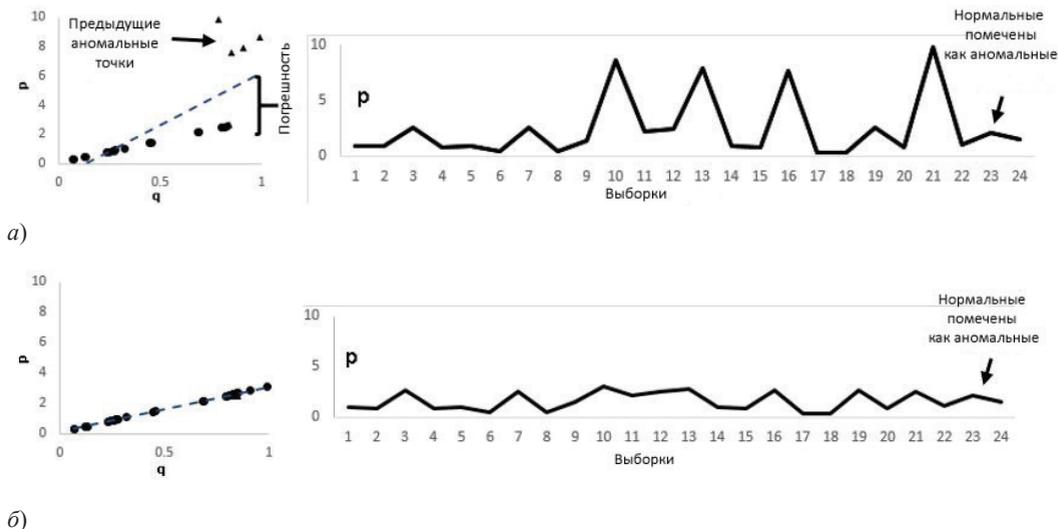


Рис. 4. Влияние пренебрежения предыдущими аномальными точками:

- а* – скользящее окно содержит выявленные аномальные точки, тогда погрешность оценивания возрастает и нормальная точка может быть помечена как аномальная;
- б* – выявленные аномальные точки были исключены из скользящего окна, что снизило погрешность оценивания, в результате нормальные точки помечены как нормальные

### Результаты экспериментов

Проведена оценка предложенного алгоритма в ходе экспериментов с использованием нескольких наборов полетных данных. В работе [18] предложен набор данных, полученных для исследовательских целей на имитаторе полетов FLTz НАСА. Группа НАСА по разработке систем интеллектуального управления полетами (ИУП) использовала тренажер FLTz для создания средств управления летательным аппаратом (ЛА), планирования и обнаружения сбоев во время полета [19]. Набор данных FLTz содержит сведения по 20 полетам ЛА с неподвижным крылом. ЛА был предварительно запрограммирован на полет по случайным круговым траекториям с вылетом и посадкой в международном аэропорту Сан-Франциско. Продолжительность каждого полета составляла от 15 до 40 мин, съем данных – с частотой 1 Гц. Испытания включали все этапы полета: взлет, набор высоты, крейсерский режим и снижение. Каждый полет характеризуется 36 параметрами, для проведения экспериментов в рамках настоящей работы было выбрано 18 из них (табл. 1). Значения других параметров были либо постоянными, либо нулевыми или совпадали со значениями, например, команд, подаваемых на правую и левую половины руля высоты. Выбранные 18 параметров будут зависимыми. Первые 14 параметров – целевые и относятся к показаниям датчиков, а остальные четыре – к командам. Были получены  $18 \times 14 = 252$  пары, количество которых было сокращено до 238 за счет удаления тех пар, где зависимый параметр совпадал с целевым (например, тангаж – тангаж или скорость – скорость), поскольку нас интересуют только контекстные неисправности. Для проверки нашего метода была разработана специальная программа на основе языка C# и библиотек Accord.net. Кроме того, для сравнения результатов были задействованы и другие алгоритмы обнаружения аномалий –  $k$ -средних и One-Class SVM. Алгоритм  $k$ -средних позволяет классифицировать име-

ющиеся данные по  $k$  заданным категориям. Чтобы использовать его для обнаружения аномалий, значение  $k$  принимается равным единице, что соответствует нормальному классу [20]. Кроме того, алгоритм  $k$ -средних изучает центроид кластера по обучающим данным, и для каждого примера вычисляется оценка аномалии по расстоянию до кластера.

Таблица 1

Параметры набора данных БПЛА

№	Параметр	Диапазон	Ед. изм.
1	Тангаж	$[-0,3; 0,3]$	<i>рад</i>
2	Скорость воздушного потока	$[50, 300]$	<i>м/с</i>
3	Скорость	$[100-500]$	<i>м/с</i>
4	Боковая скорость	$[-10-10]$	<i>м/с</i>
5	Вертикальная скорость	$[-20-100]$	<i>м/с</i>
6	Угловая скорость крена	$[-0,15; 0,15]$	<i>рад/с</i>
7	Угловая скорость тангажа	$[-0,05; 0,05]$	<i>рад/с</i>
8	Угловая скорость рыскания	$[-0,06; 0,06]$	<i>рад/с</i>
9	Продольное ускорение	$[-10, 15]$	<i>м/с<sup>2</sup></i>
10	Боковое ускорение	$[-8, 8]$	<i>м/с<sup>2</sup></i>
11	Вертик. ускорение	$[-40, 40]$	<i>м/с<sup>2</sup></i>
12	Угловое ускорение крена	$[-0,3; 0,3]$	<i>рад/с<sup>2</sup></i>
13	Угловое ускорение тангажа	$[-0,15; 0,15]$	<i>рад/с<sup>2</sup></i>
14	Ускорение по углу рыскания	$[-0,03; 0,03]$	<i>рад/с<sup>2</sup></i>
15	Команда, подаваемая на левый элерон	$[-10, 4]$	—
16	Команда, подаваемая на правый элерон	$[-10, 4]$	—
17	Команда, подаваемая на руль высоты	$[-25, 25]$	—
18	Команда, подаваемая на руль поворота	$[-6, 6]$	—

Алгоритм One-Class SVM изучает область (границу), содержащую примеры обучающих данных. Алгоритм определяет принадлежность каждого тестового примера к изученной области. Если тестовый пример попадает в изученную область, он обозначается как нормальный, а если нет – как аномальный [2]. Все реализованные алгоритмы были проверены для каждой пары параметров с использованием одной и той же платформы и при одних и тех же условиях (одинаковые наборы данных и входные данные на каждом временном шаге). Для обучения и тестирования было построено несколько комбинаций исходных наборов данных FLTz. Обучающий набор данных (28510 строк) использовался во всех экспериментах и для всех алгоритмов. Кроме того, было сформировано шесть тестовых наборов данных. Первый тестовый набор данных DS0 не содержал аномалий и состоял из 1108 строк. Набор DS0 применялся при оценивании методом DLR. Остальные тестовые наборы данных DS1–5 были одинакового размера (4539 строк). Поскольку в наборе данных FLTz нет известных неисправностей, в значения различных параметров (тангаж, угловая скорость тангажа, угловая скорость крена, угловая скорость рыскания и скорость воздушного потока) всех наборов данных, кроме DS0, были введены случайные сбои, связанные с уходом датчика. Это означает, что датчик отклоняется от истинного значения на постоянную величину. В табл. 2 представлены тестовые наборы данных и диапазоны значений введенных сбоев из-за ухода датчиков для каждого целевого параметра.

**Определение эффективности оценивания методом DLR.** Чтобы определить эффективность оценивания методом DLR, использовалась средняя абсолютная погрешность (САП), заданная формулой

$$MAE = \frac{1}{n} \sum_{t=1}^n |x_p^t - \hat{x}_p^t|, \quad (5)$$

где  $n$  – размер набора данных [21]. Расчет значений  $\hat{x}_p^t$  выполнялся по формуле (3), а размер скользящего окна для каждой пары параметров составлял 30 точек. Набор данных DS0 использовался для оценивания целевых параметров с помощью DLR. В табл. 3 показаны значения САП для каждой пары параметров. Как видно из табл. 3, все значения САП находятся в допустимых пределах.

Таблица 2.

Тестовые наборы данных

Набор данных	Размер набора данных	Кол-во сбоев	Неисправные датчики	Диапазон сбоя из-за ухода
DS0	1108	0	—	—
DS1	4539	200	Тангаж	[0,3, 0,5] рад
DS2	4539	161	Угловая скорость тангажа	[0,07, 0,1] рад/с
DS3	4539	182	Угловая скорость крена	[0,2, 0,3] рад/с
DS4	4539	145	Угловая скорость рыскания	[0,1, 0,15] рад/с
DS5	4539	168	Скорость воздушного потока	[35, 50] м/с

Наибольшие значения САП отмечены для пар, в которых целевыми параметрами являются вертикальная скорость или скорость ( $MAE \in [1,22; 1,96]$ ). Это связано с тем, что на САП влияет большой диапазон этих параметров (см. табл. 1).

На рис. 5 представлены результаты оценивания для значений скорости ЛА и погрешности оценивания. Как видно из рисунка, диапазон значений скорости воздушного потока значительно превышает диапазон заданных сбоев. Это привело к существенной погрешности оценивания в некоторых точках, следовательно, вероятность обнаружения будет низкой, а вероятность ложной тревоги – высокой.

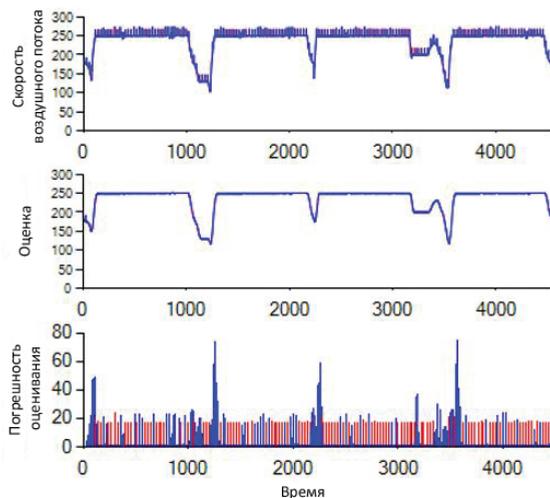


Рис. 5. Исходные значения скорости воздушного потока, оценки и погрешности оценивания при использовании DLR

Таблица 3

## Значения САП для всех пар параметров

Зависимые параметры Целевые параметры	Зависимые параметры													
	Тангаж	Воздушная скорость	Скорость	Боковая скорость	Вертикальная скорость	Угловая скорость крена	Угловая скорость тангажа	Угловая скорость рыскания	Продольное ускорение	Боковое ускорение	Вертикальное ускорение	Ускорение крена	Ускорение тангажа	Ускорение рыскания
Тангаж	–	0,73	1,53	0,76	1,61	0,02	0	0,01	0,51	0,46	0,48	0,01	0	0
Скорость воздушного потока	0,01	–	1,6	0,85	1,71	0,02	0	0,01	0,46	0,45	0,49	0,01	0	0
Скорость	0,01	0,38	–	0,85	1,52	0,02	0	0,01	0,45	0,44	0,53	0,01	0	0
Боковая скорость	0,02	0,57	1,66	–	1,91	0,02	0,01	0,01	0,54	0,44	0,5	0,02	0	0
Вертикальная скорость	0,01	0,79	1,66	0,84	–	0,03	0	0,01	0,49	0,48	0,55	0,01	0	0
Угловая скорость крена	0,02	0,56	1,66	0,41	1,89	–	0,01	0,01	0,51	0,44	0,47	0,01	0	0
Угловая скорость тангажа	0,01	0,63	1,34	0,71	1,22	0,03	–	0,01	0,54	0,48	0,54	0,01	0	0
Угловая скорость рыскания	0,01	0,53	1,24	0,8	1,33	0,03	0,02	–	0,46	0,48	0,56	0,01	0	0
Продольное ускорение	0,01	0,58	1,74	0,82	1,82	0,02	0,01	0,01	–	0,48	0,57	0,01	0	0
Боковое ускорение	0,03	0,55	1,73	0,84	1,95	0,03	0,01	0,02	0,5	–	0,43	0,01	0,01	0
Вертикальное ускорение	0,02	0,66	1,74	0,82	1,78	0,03	0,01	0,01	0,49	0,41	–	0,01	0	0
Угловое ускорение крена	0,03	0,52	1,67	0,85	1,93	0,03	0,01	0,01	0,5	0,29	0,44	–	0	0
Угловое ускорение тангажа	0,01	0,66	1,64	0,74	1,8	0,02	0,01	0,01	0,55	0,44	0,46	0,01	–	0
Ускорение по углу рыскания	0,03	0,55	1,68	0,74	1,91	0,02	0,01	0,02	0,49	0,41	0,48	0,01	0,01	–
Команда, подаваемая на левый элерон	0,02	0,6	1,6	0,37	1,96	0,01	0,01	0,01	0,51	0,47	0,45	0,01	0	0
Команда, подаваемая на правый элерон	0,02	0,57	1,6	0,44	1,93	0,01	0,01	0,01	0,5	0,47	0,45	0,01	0,01	0
Команда, подаваемая на руль высоты	0,01	0,66	1,64	0,71	1,18	0,02	0	0,01	0,38	0,51	0,57	0,01	0	0
Команда, подаваемая на руль поворота	0,03	0,53	1,66	0,82	1,91	0,02	0,01	0,01	0,49	0,41	0,46	0,01	0	0

**Показатели оценивания.** В целом результаты тестирования детекторов аномалий можно поделить на четыре группы [22]. В истинно положительных (ИП) случаях аномальное значение обозначается как аномальное, в ложноположительных (ЛП) случаях нормальное значение обозначается как аномальное, в истинно отрицательных (ИО) случаях нормальное значение обозначается как нормальное, в ложноотри-

цательных (ЛО) случаях аномальное значение обозначается как нормальное. Чтобы оценить результаты теста на обнаружение аномалий, используются такие соотношения, как вероятность обнаружения

$$\left( VO = \frac{ИП}{ИП + ЛО} \right),$$

вероятность ложных тревог

$$\left( ВЛТ = \frac{ЛП}{ЛП + ИО} \right),$$

точность

$$\left( Точность = \frac{ИП}{ИП + ЛП} \right).$$

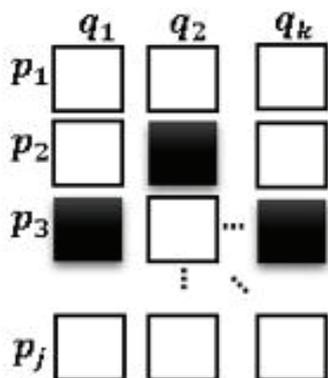


Рис. 6. Концентрация неисправностей

Эффективный алгоритм должен обеспечивать максимальную вероятность обнаружения и точность, одновременно сводя к минимуму вероятность ложных тревог. Чтобы проверить способность алгоритма описывать контексты потенциальных неисправностей, рассмотрим еще одну иллюстрацию (рис. 6). На рисунке показана матрица из черно-белых клеток, цвет каждой клетки обозначает количество обнаруженных аномальных точек среди точек каждой пары параметров. Цветовая градация клеток определяется по формуле

$$\left( Цвет = \frac{ИП + ЛП}{Число аномалий} \right).$$

Белый цвет означает, что аномалии отсутствуют, а черный – что все значения рассматриваемой пары оказались аномальными. Эффективный алгоритм показывает черный цвет только в тех клетках, где целевой параметр связан с неисправным датчиком.

**Обнаружение неисправностей.** Как показано в табл. 4, время обработки при использовании алгоритма DLR–KNN было более длительным, чем при использовании двух других алгоритмов. Для оценки скорости нового метода было измерено время обработки одного кадра, что важно при использовании в режиме реального времени. Из табл. 4 видно, что предложенный алгоритм в десять раз медленнее, чем другие известные алгоритмы. Дальнейшая работа должна быть направлена на улучшение этого показателя. Вместе с тем DLR–KNN обеспечил более точные значения, чем алгоритм *k*-средних, который характеризуется более низкой точностью (около нуля) и более высокой вероятностью ложных тревог ( $\geq 0,8$ ), что крайне неудобно. Алгоритмы SVM и DLR–KNN показали схожие результаты при обработке наборов данных DS1–DS4, но при этом с помощью алгоритма SVM не удалось найти аномалии в наборе данных DS5 из-за внезапной и существенной изменчивости значений скорости воздушного потока. На рис. 7 представлено скопление обнаруженных аномалий для каждого алгоритма при обработке набора данных DS3, аномалии были введены в его 6-й параметр (угловая скорость крена). Из рис. 7, а видно, что результаты алгоритма DLR–KNN лучше, чем у других алгоритмов, поскольку он точно определил неисправный датчик и его контекст. Из рис. 7, б видно, что алгоритм *k*-средних не смог выявить неисправный датчик и его контекст – об этом говорит черный цвет большинства клеток (см. подраздел «Показатели оценивания»).

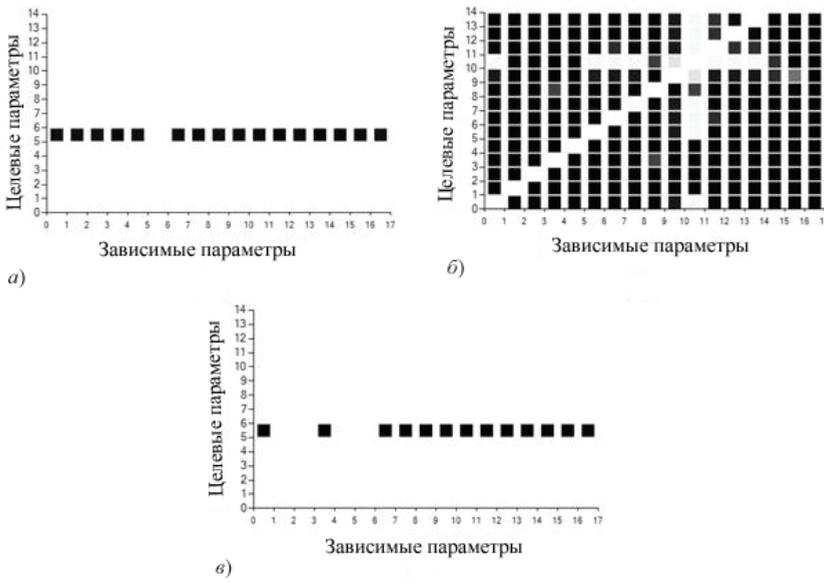


Рис. 7. Скопление неисправностей в наборе данных DS3.  
 Потенциальные сбои датчика угловой скорости крена:  
 а – скопление неисправностей для DLR–KNN;  
 б – скопление неисправностей для алгоритма *k*-средних;  
 в – скопление неисправностей для One-Class SVM

Т а б л и ц а 4

**Результаты использования алгоритмов DLR–KNN, *k*-средних и One-Class SVM**

Алгоритм	Набор данных	Время обработки кадра, мс	ВО	ВЛТ	Точность
DLR–KNN	DS1	2,29	0,99	0,006	0,88
	DS2	2,37	0,994	0,007	0,842
	DS3	2,24	0,929	0,006	0,858
	DS4	2,36	0,972	0,007	0,829
	DS5	2,37	0,601	0,017	0,574
Алгоритм <i>k</i> -средних	DS1	0,26	1	0,803	0,054
	DS2	0,26	1	0,84	0,042
	DS3	0,27	1	0,837	0,048
	DS4	0,26	0,821	0,832	0,032
	DS5	0,34	0,833	0,831	0,037
One Class SVM	DS1	0,25	1	0,001	0,976
	DS2	0,26	1	0	0,988
	DS3	0,26	1	0,003	0,938
	DS4	0,26	1	0,003	0,924
	DS5	0,29	0	0	0

**Заключение**

Функция обнаружения неисправностей является неотъемлемой частью таких сложных систем как БПЛА, поскольку она обеспечивает их безопасность. В статье предложен новый практический подход к обнаружению контекстных аномалий с помощью линейных соотношений между различными парами параметров. В этой методике такие соотношения используются для оценивания значений целевых параметров с последующей классификацией значений погрешности оценивания с по-

мощью алгоритма K-NN. Классифицированные значения погрешности оценивания используются для обозначения точек в качестве потенциальных контекстных неисправностей. Новый метод был сопоставлен с другими популярными алгоритмами –  $k$ -средних и One-Class SVM. Сравнение производилось по значениям вероятностей обнаружения, ложных тревог и точности. Скопления неисправностей приведены на рисунках, на которых черно-белая цветовая гамма клеток обозначает способность алгоритма обнаруживать контекстные неисправности. Результаты, полученные с помощью нового алгоритма, свидетельствуют о том, что по сравнению с алгоритмом  $k$ -средних он обладает большей эффективностью во всех состояниях, а по сравнению с One-Class SVM – сходной, исключая случаи, когда диапазон значений параметра значительно превышает диапазон неисправностей. В этих ситуациях новый алгоритм, в отличие от двух других, успешно справился с решением задачи. Дальнейшая работа будет направлена на значительное усовершенствование алгоритма с точки зрения его скорости. Для повышения его эффективности и точности оценивания можно задействовать более совершенные классификаторы, например LSTM (long-short memory networks – сети с долгой краткосрочной памятью) и дерево принятия решений.

### Благодарность

Работа выполнена при финансовой поддержке Высшего института прикладных наук и технологий (Дамаск, Сирия). Авторы выражают благодарность рецензентам за их полезные комментарии.

### ЛИТЕРАТУРА

1. **Colomina, I., Molina, P.**, Unmanned aerial systems for photogrammetry and remote sensing: A review, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2014, vol. 92, pp. 79–97. DOI:10,1016/j.isprsjprs.2014.02.013.
2. **Chandola, V., Banerjee, A., and Kumar, V.**, Anomaly detection: A survey, *ACM Comput. Surv.*, 2009, vol. 41, pp. 1–58. DOI:10,1145/1541880,1541882.
3. **Renckens, I.**, Automatic detection of suspicious behaviour, *Master Thesis*, 2014.
4. **Sun, R., Cheng, Q., Wang, G., and Ochieng, W.Y.**, A novel online data-driven algorithm for detecting UAV navigation sensor faults, *Sensors*, 2017, vol. 17, no. 10, p. 2243. doi:10,3390/s17102243.
5. **Ding, X., Li, Y., Belatreche, A., and Maguire, L.P.**, An experimental evaluation of novelty detection methods, *Neurocomputing*, 2014, vol. 135, pp. 313–327. DOI: 10,1016/j.neucom.2013.12.002.
6. **Pasillas-Díaz, J.R., Ratté, S.**, An unsupervised approach for combining scores of outlier detection techniques, based on similarity measures, *Electron. Notes Theor. Comput. Sci.*, 2016, vol. 329, pp. 61–77. DOI:10,1016/j.entcs.2016.12.005.
7. **Khalastchi, E., Kalech, M., Kaminka, G.A., and Lin, R.**, Online data-driven anomaly detection in autonomous robots, *Knowledge and Information Systems*, 2015, vol. 43, pp. 657–688. DOI: 10,1007/s10115-014-0754-y.
8. **Cork, L., Walker, R.**, Sensor fault detection for UAVs using a nonlinear dynamic model and the IMM-UKF algorithm, *IEEE Information, Decision and Control*, 2007, pp. 230–235.
9. **Bu, J., Sun, R., Bai, H., Xu, R., Xie, F., Zhang, Y., and Ochieng, W.Y.**, Integrated method for the UAV navigation sensor anomaly detection, *IET Radar, Sonar & Navigation*, 2017, vol. 11, pp. 847–853. DOI:10,1049/iet-rsn.2016.0427.
10. **Lin, R., Khalastchi, E., and Kaminka, G.A.**, Detecting anomalies in unmanned vehicles using the Mahalanobis distance, *Proc. IEEE International Conference on Robotics and Automation*, 2010, pp. 3038–3044. DOI:10,1109/ROBOT.2010,5509781.

11. **Khalastchi, E., Kaminka, G.A., Kalech, M., and Lin, R.,** Online anomaly detection in unmanned vehicles, *Proc. 10<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems*, 2011, vol. 1, pp. 115–122.
12. **Pokrajac, D., Latecki, L.J., and Lazarevic, A.,** Incremental local outlier detection for data streams, *Proc. IEEE Symposium on Computational Intelligence and Data Mining*, 2007, pp. 504–515. DOI:10.1109/CIDM.2007.368917.
13. **Paffenroth, R., Kay, K., and Servi, L.,** Robust PCA for anomaly detection in cyber networks, ArXiv, 2018, <https://arxiv.org/pdf/1801.01571.pdf>.
14. **Yong, D., Yaqing, X., Yuanpeng, Z., Yu, P., and Datong, L.,** Unmanned aerial vehicle sensor data anomaly detection using kernel principal component analysis, *Proc. IEEE 13<sup>th</sup> International Conference on Electronic Measurement and Instruments*, 2017, pp. 241–246.
15. **Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T.,** Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding, *Proc. 24<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, 2018, pp. 387–395.
16. **Weisberg, S.,** *Applied Linear Regression*, Hoboken, NJ: Wiley-Interscience, 2005.
17. **Ullah, I., Fayaz, M., and Kim, D.,** Improving accuracy of the Kalman filter algorithm in dynamic conditions using ANN-based learning module, *Symmetry*, 2019, vol. 11, no. 1, p. 94. DOI:10.3390/sym11010094.
18. **Oza, N.,** FLTz flight simulator, <https://c3.ndc.nasa.gov/dashlink/resources/294/>
19. **Chu, E., Gorinevsky, D., and Boyd, S.P.,** Detecting aircraft anomalies cruise flight data, *Proc. AIAA Infotech Aerospace Conference*, Atlanta, GA, 2010, pp. 1–15.
20. **Le, V.-H., Kim, S.-R.,** K-strings algorithm, a new approach based on Kmeans, *Proc. 2015 Conference on Research in Adaptive and Convergent Systems (RACS)*, 2015, pp. 15–20.
21. **Ouyang, Q., Lu, W.,** Monthly rainfall forecasting using echo state networks coupled with data preprocessing methods, *Water Resources Management*, 2018, vol. 32, pp. 659–674. DOI:10.1007/s11269-017-1832-1.
22. **Karami, A., Guerrero-Zapata, M.,** A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks, *Neurocomputing*, 2015, vol. 149, pp. 1253–1269. DOI:10.1016/j.neucom.2014.08.070.

---

**Alos, A., Dahrouj, Z.** (Department of Informatics, Higher Institute for Applied Sciences and Technology, Damascus, Syria)

Detecting Contextual Faults in Unmanned Aerial Vehicles Using Dynamic Linear Regression and K-Nearest Neighbour Classifier, *Giroskopiya i Navigatsiya*, 2020, vol. 28, no. 1 (108), pp. 66–80.

**Abstract.** Unmanned aerial vehicle (UAV) is a complex system. Its design involves control, aerodynamics, and communication systems. We use the complex linear relationships among UAV attributes (sensor readings, and commands) to propose a new technique to detect contextual faults. The contextual faults mean that a defective sensor shows invalid values concerning the context of other attributes. The proposed approach depends on estimating the values of a focused attribute using dynamic linear regression. Next, it calculates the estimation error at each time step. The values of the estimation error are classified using K-NN (Nearest Neighbour) classifier into two classes (Normal, Abnormal). The abnormal points are flagged as potential faults. Moreover, comparison with other algorithms (K-Means and One-Class SVM) is made. The proposed approach showed better results in most of the cases.

**Key words:** UAV, flight, linear regression, anomaly detection, abnormal, K-NN, classification.

Материал поступил 08.10.2019