



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
IFCE CAMPUS ARACATI
COORDENADORIA DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS SANTIAGO NASCIMENTO FERREIRA

ANÁLISE DE UM MÉTODO ONE-CLASS PARA DETECÇÃO DE
ANOMALIAS EM VÍDEOS DE SEGURANÇA

ARACATI-CE
2025

CARLOS SANTIAGO NASCIMENTO FERREIRA

ANÁLISE DE UM MÉTODO ONE-CLASS PARA DETECÇÃO DE ANOMALIAS
EM VÍDEOS DE SEGURANÇA

Trabalho de Conclusão de Curso (TCC)
apresentado ao curso de Bacharelado em
Ciência da Computação do Instituto Fede-
ral de Educação, Ciência e Tecnologia do
Ceará - IFCE - Campus Aracati, como re-
quisito parcial para obtenção do Título de
Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Silas Santiago Lopes
Pereira

Aracati-CE
2025

Dados Internacionais de Catalogação na Publicação

Instituto Federal do Ceará - IFCE

Sistema de Bibliotecas - SIBI

Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

FERREIRA, CARLOS SANTIAGO NASCIMENTO.

Análise de um método One-Class para Detecção de Anomalias em Vídeos de Segurança
/ CARLOS SANTIAGO NASCIMENTO FERREIRA. - 2025.

49 f. : il. color.

Trabalho de Conclusão de Curso (graduação) - Instituto Federal do Ceará, Bacharelado
em Ciência da Computação, Campus Aracati, 2025.

Orientação: Prof. Dr. Silas Santiago Lopes Pereira.

1. Detecção de Anomalias em Vídeos. 2. One-Class SVM (OCSVM). 3. Inflated 3D
Convolutional Neural Network (I3D). 4. Otimização de Hiperparâmetros. I. Título.

CARLOS SANTIAGO NASCIMENTO FERREIRA

ANÁLISE DE UM MÉTODO ONE-CLASS PARA DETECÇÃO DE ANOMALIAS
EM VÍDEOS DE SEGURANÇA

Trabalho de Conclusão de Curso (TCC)
apresentado ao curso de Bacharelado em
Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE - Campus Aracati, como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Aprovada em 26 de outubro de 2025

BANCA EXAMINADORA

Prof. Dr. Silas Santiago Lopes Pereira (Orientador)
IFCE

Prof. Me. Gabriel Jonas da Silva Duarte
IFCE

Prof. Dr. Mário Wedney de Lima Moreira
IFCE

AGRADECIMENTOS

A instituição me trouxe amigos que levo em meu coração, especialmente minha turma. O tempo irá nos separar — ele é inevitável. Mas todos vocês sempre estarão em minha memória com muito carinho, e o sentimento que permanece é de que criamos uma relação tão autêntica dentro e fora de sala, como se fôssemos uma segunda família. Não é à toa que chamo-os de primos.

Meus sinceros agradecimentos a todos os meus professores, principalmente aqueles que me demandaram mais esforço em suas disciplinas, pois foram vitais para que eu pudesse me desenvolver academicamente.

Agradeço imensamente ao meu orientador e xará Dr. Silas Santiago, que confiou em mim, e reservou seu tempo para me guiar desde os fundamentos mais básicos. O senhor me instruiu na busca do primor técnico, o que me inspirou a perseguir sempre o melhor em todas as etapas deste trabalho.

Agradeço aos professores da banca avaliadora Me. Gabriel Jonas da Silva Duarte e Dr. Mário Wedney de Lima Moreira, por aceitarem avaliar meu trabalho, onde possuo um grande respeito por ambos e carrego a certeza de que suas contribuições são valiosas para o presente estudo.

Carlos Alberto (Beto), muito obrigado por sempre me exigir o estudo em primeiro lugar, mesmo em momentos de aperto financeiro, o senhor nunca cogitou me deixar perder o foco na faculdade. Jeuzeline Muniz (Pêpeu), muito obrigado por sempre me motivar e estar presente comigo. A senhora nunca me deixou desistir, mesmo nos momentos em que eu mesmo desistiria de mim. Não há palavras para descrever o quão sou grato e privilegiado por ter os senhores como meus pais.

Agradeço à minha irmã Ana Thays, que sempre me defendeu na comparação entre irmãos e nunca me negou ajuda quando eu precisava acessar o computador remotamente fora de casa para executar algo do estudo.

Minha vó Angelita (Neném), muito obrigado por sempre me prover apoio quando precisei. Mesmo sem estudo, é uma das pessoas mais sábias que conheço e sempre me aconselhou bem sobre como trilhar meu futuro.

Minha querida namorada Késsya, obrigado por estar comigo por todo este percurso, me motivar, e ser compreensiva em todos os momentos em que o TCC apertou. Se a vida permitir, quero permanecer onde me sinto à vontade: ao seu lado.

Por fim, agradeço ao restante de minha família, meus primos, que tenho como irmãos, e minhas tias e tios. Família não é apenas nosso sangue; agradeço especialmente ao Sr. José Alves (Seu Dedé) por ter me fornecido internet de graça por todos estes anos, sempre a justificar que era uma forma de contribuir para meus estudos.

Dedico este trabalho ao meu avô materno
José da Silva Muniz (Zé Muniz), que
sempre foi uma fonte interminável de
inspiração em minha vida.

RESUMO

Detectar anomalias em vídeos é uma tarefa árdua, uma vez que esses eventos possuem poucas ocorrências em longas sequências de situações normais e são fortemente dependentes do contexto. Empregar anotações totais ou parciais em vídeos para o reconhecimento de tarefas específicas não é sempre viável, dado que rótulos em nível de *frame* ou vídeo podem não estar disponíveis. Este trabalho investiga o *One-Class SVM* para detecção de anomalias em vídeos de vigilância e fornece uma análise comparativa entre o método e outras abordagens avançadas para detecção de anomalias. Para otimizar o desempenho do algoritmo, é realizado um *Grid Search* para encontrar melhores hiperparâmetros. O estudo usa atributos I3D-RGB dos vídeos extraídos do *dataset ShanghaiTech*. Nos experimentos realizados sobre o conjunto de teste, a versão otimizada do modelo desempenhou com AUC de 68,41%, 0,33% de FPR e 1,52% de F1-Score. Em comparação com o estado da arte, o método I3D-RGB OCSVM obteve resultados medianos e foi superado por outros modelos mais robustos, como o *DeepSVDD* e métodos baseados em *Autoencoder*. O repositório do projeto pode ser encontrado em <https://github.com/santivgo/tcc-occ-vad>.

Palavras-chave: Detecção de Anomalias em Vídeos. *One-Class SVM* (OCSVM). *Inflated 3D Convolutional Neural Network* (I3D). Otimização de Hiperparâmetros.

ABSTRACT

Detecting anomalies in videos is an arduous task, since these events occur infrequently within long sequences of normal situations and are highly dependent on context. Employing full or partial annotations in videos for the recognition of specific tasks is not always feasible, given that frame-level or video-level labels may not be available. This work investigates the One-Class SVM for anomaly detection in surveillance videos and provides a comparative analysis between this method and other advanced approaches for anomaly detection. To optimize the algorithm's performance, a Grid Search is conducted to find the best hyperparameters. The study uses I3D-RGB features extracted from the ShanghaiTech dataset. In the experiments performed on the test set, the optimized version of the model achieved an AUC of 68.41%, 0.33% FPR, and 1.52% F1-Score. Compared with the state of the art, the I3D-RGB OCSVM method achieved median results and was outperformed by more robust models, such as DeepSVDD and Autoencoder-based methods. The source code is available at <https://github.com/santivgo/tcc-occ-vad>.

Keywords: Video Anomaly Detection. One-Class SVM. Inflated 3D Convolutional Neural Network. Hyperparameter Optimization

LISTA DE ILUSTRAÇÕES

Figura 1 – Métodos de aprendizado em <i>Machine Learning</i> no estado da arte	18
Figura 2 – Representação 2D do processo de <i>Grid Search</i>	20
Figura 3 – Estrutura da rede I3D	22
Figura 4 – Fluxo metodológico para o estudo	29
Figura 5 – Amostras de clipes do <i>ShanghaiTech</i>	30
Figura 6 – Fluxo de composição dos conjuntos de treino e validação	31
Figura 7 – Fluxo de organização dos <i>Ground Truths</i>	32
Figura 8 – Curva ROC dos modelos OCSVM no conjunto de validação	38
Figura 9 – Curva ROC dos modelos no conjunto de teste	40

LISTA DE TABELAS

Tabela 2 – Organização dos dados por conjunto	32
Tabela 1 – Conjuntos de valores utilizados para o <i>Grid Search</i> do OCSVM . . .	34
Tabela 2 – Comparativo entre os modelos OCSVM no conjunto de validação .	37
Tabela 3 – Desempenho de diferentes métodos OCC no conjunto de teste do <i>ShanghaiTech</i> , ordenado a partir da AUC	39
Tabela 4 – Comparativo de estudos de detecção de anomalias em vídeo que empregam o <i>ShanghaiTech</i>	41

LISTA DE QUADROS

Quadro 1 – Comparação de estudos com métodos <i>One-Class</i> para detecção de anomalias em vídeo	26
Quadro 2 – Comparação entre <i>datasets</i> para detecção de anomalias em vídeo	30
Quadro 3 – Matriz de confusão	35
Quadro 4 – Métricas de avaliação de modelos de classificação.	36
Quadro 5 – Matriz de Confusão do OCSVM otimizado	37
Quadro 6 – Matriz de Confusão do OCSVM padrão	37
Quadro 7 – Matriz de confusão para o OCSVM otimizado no conjunto de teste	39

LISTA DE ABREVIATURAS E SIGLAS

DL	<i>Deep Learning</i>
ML	<i>Machine Learning</i>
GAN	<i>Generative Adversarial Networks</i>
CNN	<i>Convolutional Neural Networks</i>
AE	<i>Autoencoder</i>
OCC	<i>One-Class Classification</i>
OCSVM	<i>One-Class Support Vector Machine</i>
VAD	<i>Video Anomaly Detection</i>
I3D	<i>Inflated 3D Convolutional Network</i>
LSTM	<i>Long Short-Term Memory</i>
RBF	<i>Radial Basis Function</i>
RGB	<i>Red, Green and Blue</i>
GT	<i>Ground Truth</i>
AUC	<i>Area Under the Curve</i>
ROC	<i>Receiver Operating Characteristic</i>
CPU	<i>Central Processing Unit</i>
GPU	<i>Graphics Processing Unit</i>
RAM	<i>Random Access Memory</i>
SSD	<i>Solid State Drive</i>
FPR	<i>Taxa de Falsos Positivos</i>
TPR	<i>Taxa de Verdadeiros Positivos</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	14
1.2	Objetivos	15
1.3	Organização do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Machine Learning	16
2.2	One-Class Classification	17
2.3	One-Class Support Vector Machine	18
2.4	Grid Search	20
2.5	Representação de atributos de vídeo	21
2.5.1	Transfer Learning	21
2.5.2	Inflated 3D Convolutional Neural Network	22
2.6	Detecção One-Class de Anomalias em Vídeos de Vigilância	23
3	TRABALHOS RELACIONADOS	24
4	METODOLOGIA	29
4.1	Aquisição do Dataset	29
4.2	Organização dos Dados	31
4.3	Modelagem	33
4.4	Seleção de Hiperparâmetros	33
4.5	Avaliação de Desempenho	34
5	RESULTADOS	37
6	CONCLUSÃO	43
	Referências	44

1 INTRODUÇÃO

1.1 Motivação

Na pesquisa *Global Advisor: Crime and Law Enforcement* realizada pela Ipsos em 2023, 34% dos entrevistados relataram perceber um aumento na criminalidade em seus bairros em relação ao ano anterior (Jackson, 2023). Jung e Wheeler (2023) estudam o impacto das câmeras de segurança e taxas de crime em Dallas (EUA). Os autores relacionam a adoção de câmeras de segurança com um aumento médio de 2% na quantidade de casos resolvidos, principalmente para comportamentos relacionados a furtos. Embora modesto, o percentual sugere que as câmeras podem contribuir para a resolução de cenas documentadas, e a sua relevância pode ir além da segurança pública, como na detecção de comportamentos anômalos diversos.

Apesar dos benefícios positivos da utilização de câmeras de vigilância, o monitoramento de forma contínua é uma tarefa exaustiva demais para ser realizada por operadores humanos, e pode ser impactada pela dificuldade de manter a concentração de forma contínua. Neste contexto, a aplicação de sistemas automatizados para identificar eventos anômalos se torna mais eficiente, porque, diferente de humanos, esses sistemas não sofrem os impactos físicos ou mentais de grandes jornadas de trabalho (Ilyas et al., 2021).

Entretanto, a definição de uma atividade anômala depende do ambiente em que ocorre a ação (Deepak et al., 2021). Por exemplo, a ação de correr pode ser comum em um parque, mas anormal em um shopping ou banco. Além disso, o termo “anomalia” envolve um campo abrangente de atividades atípicas, o que torna difícil sua categorização, uma vez que essas ocorrências são raras e aparecem em meio a longas sequências de gravações normais (Asad et al., 2022).

Nos últimos anos, algoritmos de *Machine Learning* (ML), especialmente aqueles baseados em *Deep Learning* (DL), como *Convolutional Neural Networks* (CNNs), *Autoencoders* (AEs) e *Generative Adversarial Networks* (GANs), têm se destacado na resolução de problemas complexos em visão computacional (Xia; Gao, 2021). No entanto, muitas dessas abordagens foram projetadas para cenários supervisionados, em que os modelos são treinados a partir de exemplos rotulados como anômalos ou não.

A rotulação de dados anômalos se torna inviável quando aplicada à detecção de anomalias em vídeo em cenários nos quais existem somente registros da classe normal (Seliya; Abdollah Zadeh; Khoshgoftaar, 2021). Para mitigar o problema da falta de dados rotulados para mais de uma classe, foram propostas abordagens capazes

de aprender a partir de informações não rotuladas, como a *One-Class Classification* (OCC), que usa apenas uma classe — geralmente, a classe normal — para classificar os dados em normais ou anômalos (Algarni; Ahmad; Alaa Ala'Anzy, 2024).

Com o aumento da demanda por sistemas autônomos de vigilância e a importância crescente da detecção de eventos irregulares em vídeo (Asad *et al.*, 2022), torna-se necessária uma análise sobre a eficácia de métodos OCC.

1.2 Objetivos

Este estudo investiga um método *One-Class* para a tarefa de detecção de anomalias em vídeos de vigilância em cenários com dados majoritariamente normais para treinamento.

Os objetivos específicos deste trabalho estão sumarizados a seguir:

1. Investigar o desempenho do método clássico *One-Class* SVM (OCSVM) para detecção de anomalias em vídeos e identificar suas melhores configurações no *benchmark* avaliado;
2. Treinar, validar e testar o modelo com base no *ShanghaiTech*, um *dataset* direcionado à detecção de anomalias em vídeo;
3. Comparar o desempenho do algoritmo estudado com outros métodos *One-Class* existentes.

1.3 Organização do Trabalho

No capítulo 2, são apresentados os conceitos essenciais para o entendimento deste estudo. Em seguida, o capítulo 3 revisa os trabalhos relacionados ao desenvolvimento da pesquisa e discute os pontos fortes e fracos de cada correlato. O capítulo 4 detalha a condução do estudo e aprofunda o funcionamento das técnicas utilizadas. No capítulo 5, apresenta-se a avaliação quantitativa dos resultados obtidos. Finalmente, o capítulo 6 expõe os objetivos alcançados, destaca as lacunas de pesquisa identificadas e aponta direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Para uma melhor compreensão dos aspectos técnicos deste trabalho, esta seção apresenta os conceitos fundamentais relacionados à área de detecção de anomalias em vídeo, como *Machine Learning* e seus paradigmas de aprendizado, *One-Class Classification*, *One-Class SVM*, representação de atributos, hiperparâmetros e *Grid Search*, e por fim, uma visão geral sobre os desafios na área de detecção de anomalias em vídeos.

2.1 Machine Learning

Goodfellow, Bengio e Courville (2016) definem algoritmos de aprendizado de máquina (em inglês, *Machine Learning*) como sistemas capazes de aprender a partir de dados. Essa capacidade de generalização faz com que estes algoritmos possam reconhecer padrões complexos e poder tomar decisões inteligentes baseadas em seu aprendizado, o que resulta em um modelo de *Machine Learning*, obtido pela aplicação do algoritmo sobre um *dataset* de treinamento (Kaisler et al., 2023).

A abordagem clássica de Mitchell (2013) define que um programa aprende com a experiência E em relação a uma classe de tarefas T e à medida de desempenho P , se o seu desempenho em tarefas de T , conforme mensurado por P , melhora com a experiência E . Em um sistema de recomendação de um *e-commerce*, a tarefa T seria recomendar produtos baseados na experiência E representada pelo histórico de compras do usuário no site, em que o desempenho P seria a porcentagem das vezes que um usuário fez uma compra sugerida e a avaliou positivamente.

Com a capacidade de aprendizado, tarefas como classificação, regressão, transcrição ou detecção de anomalias são apenas alguns exemplos de problemas que podem ser resolvidos com *Machine Learning* (Goodfellow; Bengio; Courville, 2016). Han, Kamber e Pei (2012) conceituam os paradigmas clássicos para problemas de aprendizado de máquina, a saber:

- **Aprendizagem supervisionada:** Nos algoritmos de aprendizagem supervisionada, o usuário insere um *dataset* (conjunto de dados), em que cada amostra dentro deste conjunto tem um rótulo associado a ela como entrada para o modelo. Por exemplo, o algoritmo é treinado a partir de um *benchmark* que contém fotos de biscoitos. A partir das próximas entradas, o algoritmo deve ser capaz de classificar uma foto de entrada qualquer como biscoito ou não biscoito.

- **Aprendizagem não-supervisionada:** Diferente do paradigma anterior, neste, as amostras do conjunto de entrada não são rotuladas. Ele é comumente empregado em tarefas de agrupamento (*clustering*, do inglês), cujo objetivo é identificar padrões e similaridades em grandes conjuntos de dados, o que permite a classificação do conjunto em subconjuntos de classes.
- **Aprendizagem semissupervisionada:** É uma classe de técnicas de *Machine Learning* que faz uso de dados rotulados e não rotulados durante o treinamento de um modelo. Os exemplos com rótulos são usados para aprender os modelos de classe, e os não rotulados são usados para refinar os limites entre as classes.

Tarefas de classificação, como as realizadas por métodos de aprendizado supervisionado e semissupervisionado, funcionam bem se há um *dataset* com dados rotulados. No entanto, em muitos cenários do mundo real, amostras rotuladas estão disponíveis apenas para uma das classes e classificar corretamente a classe alvo torna-se muito desafiador e, às vezes, inviável com classificadores binários tradicionais (Seliya; Abdollah Zadeh; Khoshgoftaar, 2021).

Considere um algoritmo supervisionado de classificação utilizado em uma fábrica de talheres, na qual os itens produzidos podem ser classificados apenas como colheres, garfos ou facas. Para uma amostra pertencente ao domínio destes talheres, é possível atribuí-la corretamente a uma das três classes, uma vez que há uma definição precisa das categorias que compõem o problema. Entretanto, a detecção de talheres defeituosos apresenta maior complexidade, dado que seria inviável prever e rotular todos os tipos possíveis de defeitos visuais, que podem variar desde uma haste levemente torta até um talher completamente deformado.

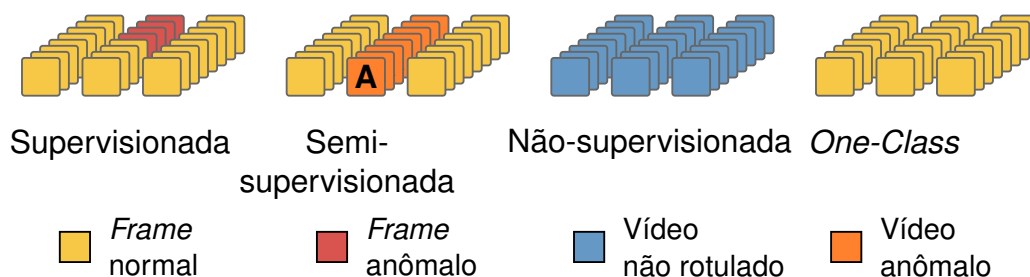
Neste contexto, uma abordagem mais adequada seria a classificação *One-Class*, de forma que o modelo aprenderia a reconhecer os itens de aço (classe normal) e trataria qualquer entrada que não pertencesse à classe normal como anomalia (Han; Kamber; Pei, 2012).

2.2 One-Class Classification

A *One-Class Classification* é uma abordagem para tarefas de *Machine Learning* que aprende apenas com dados de uma das classes. Em problemas de classificação *One-Class*, o *benchmark* é composto predominantemente por instâncias da classe negativa, que representam o comportamento normal e recorrente observado no treinamento, enquanto a classe positiva, ou anômala, corresponde à classe de interesse, caracterizada por um número minúsculo de instâncias em relação à classe

negativa (Seliya; Abdollah Zadeh; Khoshgoftaar, 2021). Um comparativo entre a abordagem OCC e os outros três paradigmas de *Machine Learning* é apresentado na Figura 1.

Figura 1 – Métodos de aprendizado em *Machine Learning* no estado da arte



Fonte: Adaptada de Zaheer, Mahmood, Khan *et al.* (2022).

A depender da literatura, a taxonomia da abordagem de aprendizado com dados de uma única classe pode ser diferente, por vezes descrita como aprendizado não supervisionado como em Ruff *et al.* (2018), ou classificação *One-Class* em Tian *et al.* (2021). Para este estudo, adota-se essa última denominação para conceituar a abordagem.

2.3 One-Class Support Vector Machine

Proposto por Schölkopf *et al.* (2001), o *One-Class Support Vector Machine* é um método para detecção de anomalias baseado na abordagem OCC. Com o objetivo de distinguir a classe positiva de qualquer outro dado no espaço, é criado um hiperplano que separa os dados da origem, que se posiciona o mais distante possível dela e isola regiões que não possuem dados, de forma que os pontos que estiverem fora dessa região sejam identificados como anomalias. Se um ponto de teste sai da região de decisão, este é classificado como anomalia (Khan; Madden, 2014). Para que essa fronteira seja definida, o algoritmo se apoia em um subconjunto particular: os vetores de suporte, que correspondem aos exemplos de treinamento mais próximos do limite de decisão e são fundamentais para definir a fronteira entre a classe normal e potenciais anomalias (Noble, 2006).

(Kittidachanan *et al.*, 2020) define os parâmetros de maior importância para o *One-Class SVM*, a seguir:

- **Contamination:** A proporção dos dados do conjunto será considerada como anômala (Masakuna *et al.*, 2025);

- **Kernel:** A função matemática usada para transformar os dados em um espaço de maior dimensão, para tornar mais fácil encontrar ou separar padrões não lineares (Hofmann; Schölkopf; Smola, 2008);
- **Gamma:** Coeficiente da função. Controla a influência de cada ponto de treino (Chen et al., 2025). Em um *gamma* alto, cada ponto tem uma região de influência pequena, com a tendência de memorizar os dados de treinamento (*overfitting*). Já com um *gamma* muito baixo, o modelo pode generalizar todos os dados como normais, mesmo aqueles que são de fato anômalos;
- **Nu:** Representado matematicamente por ν , o parâmetro *nu* determina um limite superior para a fração de erros de treinamento e um limite inferior para a fração de vetores de suporte (Fujita; Matsukawa; Suzuki, 2020).

De acordo com Ilyas et al. (2021), dado um *dataset* de treinamento $S = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, o problema *One-Class* é formulado como o seguinte programa quadrático:

$$\begin{aligned} \min_{\mathbf{w}, \rho} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu n} \sum_{i=1}^n \xi_i - \rho \\ \text{sujeito a} \quad & \mathbf{w} \cdot \phi(\mathbf{x}_i) \geq \rho - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \dots, n. \end{aligned} \tag{2.1}$$

onde $\nu = 0,5$, \mathbf{w} é o vetor de pesos, ρ é o deslocamento, ξ_i é a variável de folga e $\phi(\cdot)$ é a função de projeção tal que

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j).$$

Com o uso da função de base radial, tem-se:

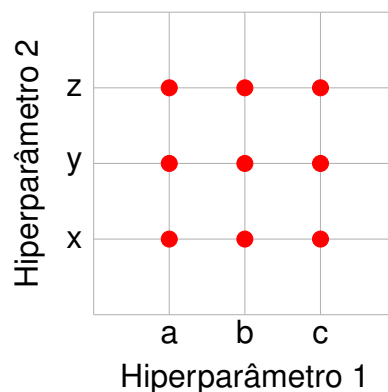
$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right). \tag{2.2}$$

Finalmente, a função de decisão é dada por:

$$f(\mathbf{x}_t) = \text{sign}(\mathbf{w} \cdot \phi(\mathbf{x}_t) - \rho). \tag{2.3}$$

Onde, $\text{sign}(\cdot)$ é a função sinal.

Algoritmos baseados no *One-Class* SVM convencional necessitam de uma quantidade significativa de memória e são computacionalmente caros à medida que o tamanho do conjunto de treinamento aumenta, com uma demanda de recursos que escala proporcionalmente ao número de amostras de treinamento (Yang; Kpotufe; Feamster, 2021).

Figura 2 – Representação 2D do processo de *Grid Search*

Fonte: Adaptada de [Liashchynskyi e Liashchynskyi \(2019\)](#).

Mesmo com a vantajosa abordagem de classificar anomalias a partir de uma única classe, o método ainda possui limitações práticas. Quando aplicado a um número limitado de amostras, pode ter um desempenho insatisfatório. Além disso, o OCSVM é bastante sensível à seleção de hiperparâmetros, o que torna sua otimização difícil ([Thomas; Feuillard; Gramfort, 2015](#)).

2.4 Grid Search

Hiperparâmetros estão presentes na criação de modelos e exercem um papel crucial em seu desempenho. Eles possibilitam ajustes que podem influenciar a complexidade, comportamento e velocidade de treinamento. Portanto, a escolha de um único parâmetro ruim pode comprometer a performance geral e impedir que o modelo atinja resultados satisfatórios ([Bischl et al., 2023](#)). Criar e treinar um modelo com sua configuração padrão, sem quaisquer otimizações ou ajustes, limita o aproveitamento total de seu potencial.

O *Grid Search* é uma técnica usada para encontrar uma combinação de parâmetros que melhore o desempenho do modelo ([Shams et al., 2023](#)), com uma busca exaustiva sobre todas as combinações de hiperparâmetros definidos previamente. Após encontrar todas as possíveis combinações, o modelo é treinado e avaliado a partir do conjunto de validação para cada configuração. A melhor combinação será aquela com maior performance no conjunto a partir de alguma métrica para avaliar o modelo, como, por exemplo, a maior AUC. Uma representação 2D do método pode ser encontrada na Figura 2, em que cada ponto em destaque indica uma combinação entre os parâmetros.

A maior vantagem do *Grid Search* é que, ao testar todas as combinações, a técnica garante que nenhuma combinação de hiperparâmetros será perdida, o que

ajuda a encontrar o modelo mais performático. Entretanto, a técnica também possui limitações. Uma desvantagem é o custo computacional, principalmente quando a pesquisa gira em torno de uma grande quantidade de hiperparâmetros e extensos intervalos de valores possíveis para cada um deles. Nessas situações, o espaço de busca pode rapidamente se tornar grande demais para ser explorado de forma exaustiva, o que resulta em longos tempos de treinamento (Leal et al., 2023).

2.5 Representação de atributos de vídeo

Ao trabalhar com dados não numéricos — como palavras, imagens, vídeos — é necessário mensurá-los numericamente antes de aplicar qualquer modelo de *Machine Learning*. O próprio fluxo de construção de um algoritmo de ML envolve não só a escolha do modelo, mas também a seleção de quais características são importantes para o problema. O uso excessivo de atributos pode tornar o modelo mais complexo e custoso para treinar; por outro lado, um conjunto reduzido de atributos pode comprometer o desempenho da tarefa (Zheng; Casari, 2018).

No estado da arte, métodos baseados em *Deep Learning* são os mais utilizados na área de detecção de anomalias em vídeo. O uso destes algoritmos permite a extração automática de atributos complexos dos dados, o que concede ao modelo a capacidade de detectar padrões relevantes (Samaila et al., 2024). Entre as arquiteturas mais usadas na extração dos atributos, destacam-se o *Autoencoder* (AE), *Generative Adversarial Networks* (GAN), *Convolutional Neural Networks* (CNN) e *Long-Short Term Memory* (LSTM).

2.5.1 Transfer Learning

Se as representações de atributos forem aprendidas eficientemente pelo modelo, é possível transferir este aprendizado entre cenários semelhantes. Essa estratégia é comum no aprendizado supervisionado, no qual a entrada permanece a mesma, mas o alvo pode ser de natureza diferente (Goodfellow; Bengio; Courville, 2016). Por exemplo, é possível aprender sobre um conjunto de categorias visuais, como gatos e cachorros, no primeiro cenário, e depois aprender sobre um conjunto diferente de categorias visuais, como formigas e vespas. Se houver significativamente mais dados no primeiro cenário do que no segundo, isso pode ajudar a aprender representações úteis para generalizar rapidamente a partir de apenas alguns exemplos do segundo cenário.

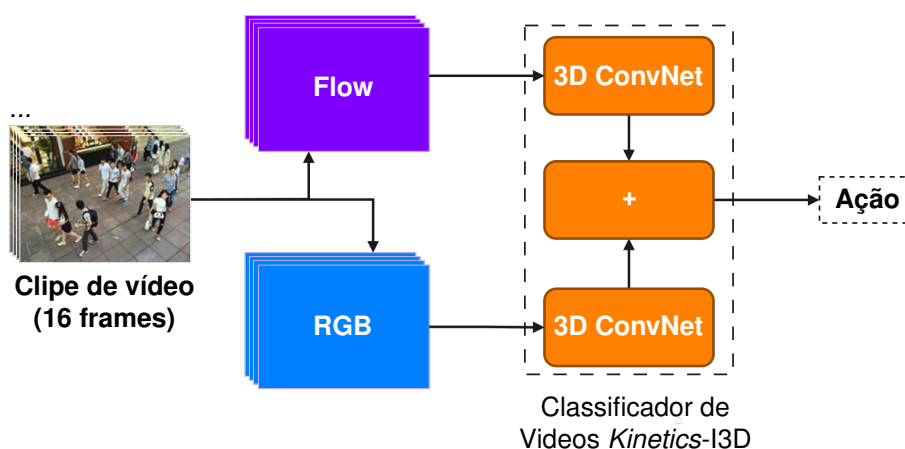
2.5.2 Inflated 3D Convolutional Neural Network

Introduzidos em Carreira e Zisserman (2017), os atributos I3D são representações de características baseadas em uma arquitetura na qual os dados são considerados em três dimensões: altura, largura e tempo. A *Inflated 3D Convolutional Neural Network* é uma rede neural profunda de extração de características, pré-treinada no *Kinetics*¹ (Kay et al., 2017), um *benchmark* para reconhecimento de ações humanas em vídeos que permite a reutilização dos atributos aprendidos para a solução de outros problemas de análise de vídeo por meio de *Transfer Learning*. A rede é capaz de utilizar tanto a aparência quanto a informação de movimento dos vídeos, que são divididos em cliques não sobrepostos com 16 *frames* consecutivos cada.

A I3D é composta por duas redes convolucionais, RGB e *Flow*. A rede RGB processa os canais de cores *red*, *green* e *blue* do vídeo de entrada e realiza o processamento espaço-temporal contínuo dos *frames*. A rede *flow* usa fluxo óptico (em inglês, *Optical Flow*), que compara o *frame* atual do vídeo com o anterior e os subsequentes, de modo a extrair as diferenças que são capturadas como atributos no vídeo. Os atributos I3D correspondem às representações espaço-temporais extraídas pelas camadas convolucionais de cada rede antes da classificação final.

Com o processamento das duas redes convolucionais RGB e *Flow*, os atributos das redes convolucionais são combinados. A saída da rede I3D é, então, a predição das classes de cada *frame* do vídeo. Para um *dataset* de videovigilância, por exemplo, a entrada da rede seria um clipe de 16 *frames* e sua saída indicaria quais são anômalos e quais são normais. A Figura 3 apresenta, de forma ilustrativa, a estrutura da rede com imagens de amostra do *benchmark ShanghaiTech* como entrada.

Figura 3 – Estrutura da rede I3D



Fonte: Adaptada de Erfanian Ebadi et al. (2021).

Essa arquitetura permite a expansão de filtros de convolução e *kernels* de po-

¹ Disponível em: <https://github.com/cvdfoundation/kinetics-dataset>

olung tradicionalmente bidimensionais ($N \times N$), com apenas largura e altura, para uma estrutura tridimensional ($N \times N \times N$), que possibilita também a captura de informações temporais dos vídeos.

2.6 Detecção One-Class de Anomalias em Vídeos de Vigilância

Samaila *et al.* (2024) definem uma anomalia como a ocorrência de eventos ou comportamentos considerados incomuns, irregulares, inesperados e diferentes de padrões existentes. Sucintamente, anormalidade é qualquer atividade cuja possibilidade de acontecer é mínima. Em situações reais, definir uma anomalia depende bastante do contexto, o que torna a construção de um algoritmo genérico, que detecte qualquer tipo de anomalia, um processo laborioso.

Para detecção e localização de anomalias em vídeos, o problema tem diversos fatores, como a complexidade do ambiente (variações de iluminação, oclusão de objetos), o tipo das anomalias, densidade de multidões, dados ruidosos, natureza complexa de comportamentos humanos, configurações da câmera, variações espaço-temporais e necessidade de uma infraestrutura computacional robusta.

Métodos de classificação *One-Class* são frequentemente usados para detecção de anomalias em vídeo (Zaheer; Mahmood; Khan *et al.*, 2022). Eles se mostram úteis em contextos nos quais é difícil obter amostras de todas as classes de anomalias, apesar de ter uma grande disponibilidade de vídeos com comportamentos normais. Em sistemas de vigilância, por exemplo, é comum ter horas de gravações que contenham cenas cotidianas, mas anomalias como furtos, invasões ou comportamentos agressivos são raramente registradas.

Apesar da grande gama de ferramentas, existem muitos problemas e desafios na área. Um dos principais é o *dataset* utilizado, em que estes, por muitas vezes, podem conter um tamanho limitado de dados ou serem compostos por dados irreais. Além disso, questões como proteção de privacidade dificultam a coleta e o compartilhamento dos dados reais. Computacionalmente, destacam-se outros desafios, como a natureza ambígua das anomalias, desbalanceamento entre as classes e alta variabilidade de eventos anômalos (Samaila *et al.*, 2024).

3 TRABALHOS RELACIONADOS

Nesta seção, são apresentados cinco trabalhos com problemáticas semelhantes à deste estudo. Tais trabalhos possuem abordagens *One-Class*, sendo que quatro destes usa o OCSVM para a classificação de anomalias em vídeo. Priorizou-se a seleção de pesquisas que empregassem diferentes formas de representar os atributos. Os correlatos foram encontrados majoritariamente no *Google Acadêmico*, filtrados no escopo de 2018 a 2025.

Em *Amraee et al. (2018)*, a detecção de anomalias em cenas com multidões é realizada por meio da combinação de descritores de movimento (HOG-LBP) e atributos de movimento (HOF) para extrair informações de aparência e padrões de movimento, respectivamente, com dois *One-Class SVMs* com função RBF para a classificação das anomalias visuais e de movimento. O *pipeline* de inferência com o *framework* proposto consiste na criação de um mapa de regiões de cada *frame*, seguido da extração de regiões candidatas a anomalias. Para regiões pequenas, são extraídos apenas atributos de movimento. Em grandes regiões, são extraídos tanto suas características quanto seus descritores. Os atributos gerados pelos HOF e HOG-LBP de movimento serão a entrada para o classificador, que identifica os dados como anomalias visuais ou de movimento. O modelo foi concebido no MATLAB, por meio da biblioteca integrada LIBSVM. Os resultados experimentais mostram que o método proposto supera métodos mais avançados do estado da arte nos *datasets* UCSD Peds¹, com uma AUC de 85% no *Ped1* e 93% no *Ped2*.

Ilyas et al. (2021) propõem um método baseado em *Hybrid Deep Network*. São combinados atributos manuais com atributos profundos. São usados dois modelos de redes neurais convolucionais *ResNet101* (*He et al., 2016*) para extrair informações espaciais e temporais profundas das imagens dos vídeos, com uma suavização da distância euclidiana entre as saídas consecutivas das redes para melhorar os atributos profundos. Para as representações extraídas de forma manual, os autores computam a soma dos gradientes da diferença dos *frames* consecutivos. A classificação é feita por uma combinação de três OCSVMs; apenas um classificador é usado para rotular o *frame* como normal, mas é necessária a decisão unânime dos três para classificá-lo como anômalo. Em testes com os *benchmarks* PETS2009 (*Ferryman; Shahrokni, 2009*) e *Unusual Crowd Activity* (UMN)², o modelo atinge 94% e 99% de AUC, respectivamente, o que evidencia que a abordagem híbrida supera outras existentes que se baseiam somente em técnicas de *Machine Learning* clássica e

¹ Disponíveis em: <http://www.svcl.ucsd.edu/projects/anomaly/dataset.html>

² Disponível em: <http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi>.

consegue resultados melhores que alguns métodos de *Deep Learning*.

Deepak *et al.* (2021) sugerem dois métodos baseados no *Multi-View Representation Learning Framework* para detecção de anomalias em vídeo. O primeiro é um método de abordagem híbrida que combina atributos extraídos por um *Autoencoder* espaçotemporal 3D (3D-STAE) com atributos extraídos manualmente, baseados na correlação espaçotemporal dos gradientes. A segunda abordagem, totalmente profunda, usa atributos extraídos de dois 3D-STAEs para transformar os dados de entrada em múltiplos espaços da modalidade e combinar os atributos extraídos. Após a extração das representações latentes, um classificador OCSVM é utilizado para identificar anomalias. Por fim, com o uso da API *Keras*, os métodos foram validados por meio dos *benchmarks Avenue* (Lu; Shi; Jia, 2013), *Live Videos* (Leyva; Sanchez; Li, 2017) e *BEHAVE* (Bhatnagar *et al.*, 2022), e os métodos híbridos atingiram, respectivamente, 78,40%, 38,40% e 85,12% de acurácia, e os métodos profundos, 82,40%, 64,90% e 80,05%, com a conclusão de que os métodos híbridos e profundos produzem resultados melhores que outros métodos do estado da arte de forma consistente nos três conjuntos de dados de referência padrão.

A proposta de Astrid *et al.* (2021) é apresentar um refinamento no método *Autoencoder* aplicado a problemas de detecção de anomalias *One-Class*. Os autores introduzem o conceito de pseudoanomalias, que são dados normais manipulados para simular a distribuição de dados anômalos, com o objetivo de treinar o modelo para reconstruir somente dados normais independentemente da entrada. Os atributos usados foram extraídos pelo próprio *Autoencoder*. O trabalho detalha dois métodos de geração de pseudoanomalias, baseados em *patch* e em *skip frame*, e demonstra que essa abordagem supera os métodos baseados no algoritmo tradicional. Os experimentos realizados nos *datasets UCSD Ped2, Avenue e ShanghaiTech* resultaram em valores de AUC de 94,77%, 84,91% e 72,46%, respectivamente, para o método baseado em *patch* e 96,50%, 84,67% e 75,97% para a versão baseada em *skip frame*. Alguns detalhes metodológicos, como o uso do *PyTorch* na implementação dos modelos, não estão descritos no artigo, mas encontram-se disponíveis no repositório do projeto³. Embora não adote o mesmo método dos demais estudos correlatos, o trabalho foi incluído entre eles por também empregar o *dataset ShanghaiTech*.

O método de Pawar e Attar (2022) combina dois AEs em *pipeline*: o primeiro é convolucional e o segundo é um *sequence-to-sequence LSTM Autoencoder*, com o objetivo de extrair as representações espaciais e temporais dos vídeos, respectivamente. Para a classificação, os autores usaram o método *One-Class SVM* com uma função de base radial (RBF) para treinar o modelo com dados normais e, para testá-

³ Disponível em: <https://github.com/aseuteurideu/LearningNotToReconstructAnomalies>.

los, usaram dados anômalos. O modelo foi implementado no *TensorFlow*, projetado de maneira que ele consegue calcular a geração de anomalias com base na distribuição gaussiana e executar detecção de anomalias com base na classificação de uma classe. Nos testes, destacam-se as bases *UCSD Ped1* e *UMN*⁴ com o AUC de 88,60% e 87%, respectivamente, com uma performance significativa em *Equal Error Rate* e de tempo necessário para detecção e localização de anomalias.

Uma comparação entre os trabalhos correlatos e o presente estudo é exposta no Quadro 1.

Quadro 1 – Comparação de estudos com métodos *One-Class* para detecção de anomalias em vídeo

Estudo	Método	Ferramentas	Representação de Atributos	Datasets
Amraee et al. (2018)	HOG-LBP e OCSVM	LIBSVM	Manual	<i>UCSD Ped1</i> e <i>UCSD Ped2</i>
Ilyas et al. (2021)	ResNet101 e OCSVM	Não especificada	Híbrida	<i>PETS2009</i> e <i>UMN</i>
Deepak et al. (2021)	3D-STAE e OCSVM	<i>Keras</i>	Híbrida	<i>Avenue</i> , <i>Live Videos</i> e <i>BEHAVE</i>
Astrid et al. (2021)	Pseudoanomalias e <i>Autoencoder</i>	<i>PyTorch</i>	Profunda	<i>UCSD Ped2</i> , <i>Avenue</i> e <i>ShanghaiTech</i>
Pawar e Attar (2022)	DLP-AE e OCSVM	<i>TensorFlow</i>	Profunda	<i>UCSD Ped1</i> e <i>UMN</i>
Este estudo	I3D (RGB) e OCSVM	<i>PyOD</i>	Profunda	<i>ShanghaiTech</i>

Fonte: Elaborado pelo autor (2025).

É possível diferenciar os tipos de métodos, ferramentas, representação de atributos e *datasets*. Sendo a coluna de maior importância, a presença de diferentes combinações de métodos tem por objetivo comparar como abordagens *One-Class* desempenham através de diferentes formas de representar as características do *dataset* utilizado. A escolha de trabalhos com representações híbridas e profundas em detrimento das manuais também se justifica pelo fato de ambas serem predominantes no atual estado da arte.

Representação de atributos é referente à forma como estes são obtidos e processados para a detecção de anomalias. Eles são considerados como profundos quando extraídos por meio de uma rede de *Deep Learning*, e híbridos quando os autores combinam as características extraídas com atributos obtidos por métodos clássi-

⁴ Disponível em: https://mha.cs.umn.edu/proj_events.shtml#crowd

cos de extração manual (*handcrafted features*). Este trabalho, por sua vez, não realiza extração de características, mas utiliza um conjunto de atributos I3D previamente extraídos e disponíveis para uso da comunidade científica para o *dataset* selecionado.

Ferramentas são os recursos utilizados na implementação do modelo, como bibliotecas e *frameworks*. O uso adequado dessas ferramentas é importante, pois muitas oferecem métodos avaliativos integrados, o que facilita a análise de desempenho e a validação do modelo.

A relevância dos correlatos é notória e são essenciais para entender a importância do desenvolvimento da investigação proposta por esse trabalho. No entanto, é necessário observar que algumas das pesquisas apresentam definições metodológicas vagas. Astrid *et al.* (2021) não incluem a ferramenta usada para construção do modelo no artigo original, sendo necessário acessar o repositório e navegar pelos arquivos da implementação. Em Ilyas *et al.* (2021), não há uma citação sequer sobre qual ferramenta foi usada, e não há nenhum código disponível para acesso.

O método OCSVM, utilizado pela maioria dos correlatos, é dependente de uma boa parametrização para obter melhores resultados. Apenas dois dos correlatos detalham a configuração do modelo. Em Deepak *et al.* (2021), os hiperparâmetros de tamanho do *kernel*, número de *kernels* e *strides* são definidos empiricamente pelos autores e os valores de *kernel* são aleatórios. Apenas em Pawar e Attar (2022) é falado sobre *Grid Search*, mas de forma simplificada, com o uso de apenas dois hiperparâmetros. É possível que a carência de uma busca mais aprofundada possa vir a prejudicar o desempenho do algoritmo.

Os trabalhos apresentados empregam atributos *handcrafted* ou combinam estes com outros que foram extraídos por outros métodos de extração. Entretanto, nenhum dos estudos considerou o uso da rede I3D no contexto estritamente OCC. Por ser pré-treinada em um *benchmark* para reconhecimento de ações humanas, a rede é capaz de extrair representações mais robustas. Apesar do uso em reconhecer ações humanas, neste trabalho, a I3D é utilizada para reconhecimento de ações anômalas.

No estado da arte, o método I3D-OCSVM já foi empregado em trabalhos anteriores. Em Roitberg, Al-Halah e Stiefelhagen (2018), o *One-Class SVM* com atributos I3D é usado como *baseline* para a avaliação de desempenho. Entretanto, o propósito dos autores é apenas identificar ações humanas em vídeo. Além disso, eles usam um outro paradigma de aprendizagem, o *Zero-Shot Learning* para treinar o modelo nos *benchmarks* UCF-101 (Soomro; Zamir; Shah, 2012) e HMDB-51 (Kuehne *et al.*, 2011). O trabalho de Nayak e Chaudhari (2023) possui uma correlação maior com o presente estudo, mas ainda sim com suas diferenças. Embora também trate de detecção de anomalias em vídeo e use o *ShanghaiTech* I3D com OCSVM, os autores optam por uma abordagem semissupervisionada, a *Multiple Instance Learning* (MIL)

para treinamento do modelo com uma função de erro personalizada.

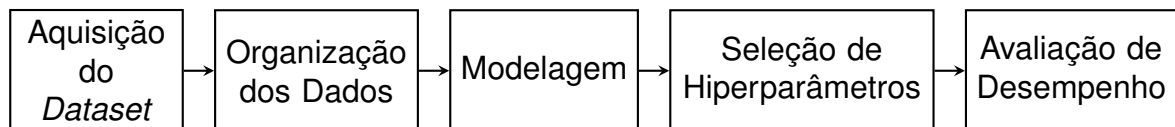
Diferentemente dos trabalhos revisados, este estudo busca analisar de forma aprofundada o método I3D-RGB OCSVM. Oferecemos um estudo sobre o impacto de cada hiperparâmetro na construção do modelo, uma busca sobre as melhores configurações e como o modelo desempenha ao ser comparado com métodos atuais e consolidados do estado da arte para detecção de anomalias em vídeo.

Apesar destes avanços, uma limitação no presente trabalho em relação aos correlatos é o uso de apenas um *benchmark* para avaliar o desempenho do modelo. O escopo da pesquisa foi delimitado pelo tempo disponível, o que inviabilizou a realização de mais experimentos. A presença de pelo menos mais um *dataset* viria a enriquecer o estudo, com um segundo resultado que consolidaria como o modelo se comporta sob diferentes condições experimentais. Mesmo com essa limitação, a contribuição metodológica do estudo não deixa de ser válida e replicável.

4 METODOLOGIA

Encontram-se nesta seção tópicos referentes à metodologia do presente estudo. O fluxo metodológico para o presente trabalho está contido na Figura 4.

Figura 4 – Fluxo metodológico para o estudo



Fonte: Elaborada pelo autor (2025).

Serão detalhadas as particularidades do *dataset* escolhido, seguido pela descrição da organização adotada, técnicas utilizadas na construção do modelo I3D-RGB *One-Class SVM*, o procedimento para a busca de melhores hiperparâmetros para o modelo e definição das métricas empregadas para a avaliação do desempenho.

4.1 Aquisição do Dataset

Conforme apresentado por Luo, Liu e Gao (2017), o *ShanghaiTech Campus Dataset* foi construído com eventos anômalos gravados no campus da universidade *ShanghaiTech*¹ em 13 locações diferentes, que incluem anormalidades encenadas como pessoas correndo e andando de bicicleta ou skate, que totalizam 437 vídeos, com aproximadamente 3,5 horas de gravações (Abdalla *et al.*, 2025). As cenas são gravadas em ambientes com condições de iluminação e ângulos de câmeras variados, o que torna o *benchmark* relativamente complexo. As divisões (em inglês, *splits*) disponibilizadas pelos autores consistem em 330 vídeos para treino e 107 vídeos com 130 eventos anormais para teste, e possuem 317.398 *frames* no total, com alguns destes presentes na Figura 5. A presença de anomalia em um vídeo é verificada a partir dos *ground truths* (GTs), que são rótulos a nível de *frame* que indicam se estes são normais ou anômalos.

Zhu, Chen e Sultani (2020) fornecem um Quadro 2 comparativo dos *datasets* mais recorrentes para a tarefa de detecção de anomalias, com destaque ao *ShanghaiTech*. Apesar do número relativamente reduzido de *frames* em comparação a outros conjuntos como *Avenue*, UMN e *UCF-Crime*, o *ShanghaiTech* oferece múltiplos ângulos de câmeras e condições de iluminações variadas, características que reforçam a sua relevância por refletirem com maior fidelidade situações reais de vigilância.

¹ Disponível em: https://svip-lab.github.io/dataset/campus_dataset.html

Figura 5 – Amostras de clipes do *ShanghaiTech*

Fonte: Luo, Liu e Gao (2017).

Quadro 2 – Comparação entre *datasets* para detecção de anomalias em vídeo

Conjunto	Nº de vídeos	Média de Frames	Anomalias de Exemplo
<i>ShanghaiTech</i>	437	317	Pessoas correndo, andando de bicicleta e skate
<i>UCF-Crime</i>	1.900	2.309	Incêndios criminosos, acidentes, assaltos, brigas
<i>UCSD Ped1</i>	70	201	Ciclistas, carrinhos de supermercado
<i>UCSD Ped2</i>	28	163	Ciclistas, carrinhos de supermercado
<i>Avenue</i>	37	839	Pessoas correndo e arremessando itens
UMN	3	1.290	Pessoas correndo
DADA	2.000	1.596	Acidentes de Trânsito
DoTA	4.677	1.150	Anomalias de Trânsito, ex. colisão
<i>Street Scene</i>	81	2509	Travessia indevida, estacionamento ilegal

Fonte: Adaptado de Zhu, Chen e Sultani (2020).

Para este estudo, foi usada a versão do *benchmark* fornecida por Wan *et al.* (2021)², que já contempla os atributos dos vídeos extraídos por meio de uma rede *Inflated 3D*. Os atributos dos vídeos estão organizados em clipes de 16 *frames*, em que cada clipe representa uma fração do vídeo original. Estudos recentes na área de detecção de anomalias em vídeos mostram a eficácia do *ShanghaiTech* com os atributos I3D extraídos pelo canal RGB, como o de Tian *et al.* (2021), em que o método proposto pontuou 97,21% de AUC no *ShanghaiTech* I3D-RGB.

A estrutura do *dataset* usada por este trabalho consiste nos arquivos de *splits* para treino, teste e *ground truths* fornecidos por Luo, Liu e Gao (2017) e os atributos I3D do canal RGB extraídos por Wan *et al.* (2021).

² Disponível em: https://github.com/wanboyang/anomaly_detection_LAD2000

4.2 Organização dos Dados

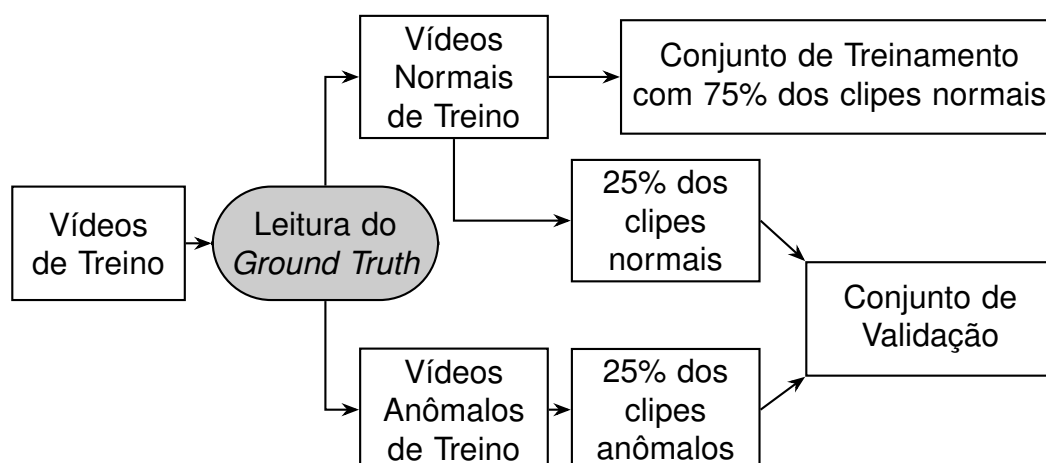
O *One-Class SVM* aprende por meio do treino com clipes dos vídeos normais; entretanto, o *ShanghaiTech* possui dados normais e anormais na sua *split* de treinamento. Além disso, não há uma partição de validação, que seria vantajosa para a etapa de *Grid Search* do modelo. Foi realizado um particionamento dos dados com o propósito de remover dados anômalos do conjunto de treinamento e criar um conjunto de validação.

A biblioteca *Pandas* (McKinney, 2010) foi usada para manipular os dados, desde a leitura dos arquivos de *splits* e dos GTs contidos no *benchmark* até a conversão dos dados em *dataframes*, estruturas tabulares próprias do *Pandas* para análise de dados computacionais.

Para remover os vídeos anômalos da *split* de treino, foi realizada uma comparação entre os identificadores (IDs) dos itens deste conjunto com vídeos rotulados como anômalos no *ground truth*, representado por um vetor de dimensão m , em que o tamanho do vetor corresponde ao número de *frames* no vídeo. Cada coluna indica se o *frame* correspondente contém ou não uma anomalia. Dessa forma, foram removidos todos os dados rotulados como anômalos dos vídeos de treino e agrupados em um conjunto separado.

Em seguida, foram realocados 25% dos clipes referentes aos vídeos normais de treino e 25% dos clipes dos vídeos anômalos de forma aleatória para compor o conjunto de validação. A partir dos vídeos normais remanescentes, foram agrupados seus respectivos clipes para a formação do conjunto de treinamento. O fluxo da organização para o conjunto de treinamento e validação está presente na Figura 6.

Figura 6 – Fluxo de composição dos conjuntos de treino e validação



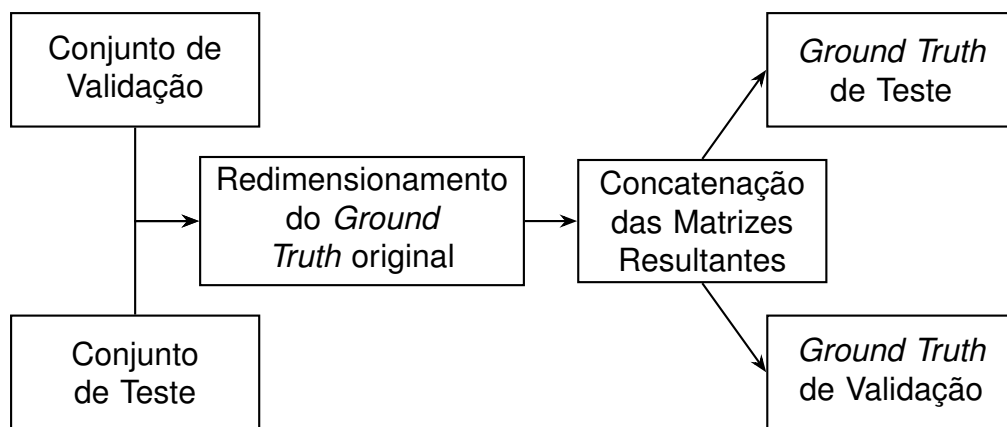
Fonte: Elaborada pelo autor (2025).

Assim, os dados foram organizados em três conjuntos, que guardam os IDs

dos respectivos cliques. Com a finalidade de verificar posteriormente se as previsões do modelo estão corretas, foram extraídos os rótulos do *ground truth* original para teste e validação e agrupados em dois vetores.

Cada vetor foi redimensionado em uma matriz bidimensional, de modo que, a cada grupo de 16 colunas — correspondentes a 16 *frames*, isto é, um clipe — formava-se uma nova linha. As matrizes resultantes ($N \times 16$) de todos os vídeos foram, então, concatenadas em duas matrizes finais, que representam os *ground truths* dos cliques presentes nos conjuntos de validação e de teste. A Figura 7 demonstra o processo de organização dos GTs.

Figura 7 – Fluxo de organização dos *Ground Truths*



Fonte: Elaborada pelo autor (2025).

A organização das partições foi finalizada com a leitura das características correspondentes a cada clipe nos vídeos do *dataset*. Cada arquivo de atributos foi adicionado ao conjunto respectivo conforme o vídeo de origem do clipe. A disposição dos dados pode ser encontrada de forma mais clara na Tabela 2.

Tabela 2 – Organização dos dados por conjunto

Categoria	Vídeos	Cliques	<i>Frames</i>
Treinamento			
Normal	132	7.181	114.896
Anômalo	-	-	-
Validação			
Normal	43	2.258	36.128
Anômalo	15	191	3.056
Teste			
Normal	155	8.242	131.872
Anômalo	44	539	8.624

Fonte: Elaborada pelo autor (2025).

Por fim, o particionamento resultou nos conjuntos de atributos I3D para treino, teste e validação, e nas matrizes de *ground truth* de teste e validação para a avaliação

das predições.

4.3 Modelagem

O classificador *One-Class SVM* foi escolhido por ser um método clássico, frequentemente usado como referência inicial (Tao *et al.*, 2024; Shi *et al.*, 2025; Roitberg; Al-Halah; Stiefelhagen, 2018). Mesmo em trabalhos recentes, métodos baseados no classificador apresentaram resultados promissores para lidar com a detecção de anomalias em vídeo, como em Yao *et al.* (2024), em que o modelo atingiu uma AUC de 84,30% com o *ShanghaiTech* em seus resultados experimentais.

O uso do *PyCaret*³ como ferramenta foi considerado no início do estudo por ser uma biblioteca amplamente utilizada no estado da arte para tarefas de *Machine Learning*. Entretanto, sua utilização foi descartada no início do estudo, uma vez que o *PyCaret* funciona como um *wrapper* da biblioteca *PyOD* — isto é, faz uso das funções desta — e não oferece uma maior personalização na definição dos hiperparâmetros dos modelos.

Com um maior suporte ao *One-Class SVM* e a outros modelos para detecção de anomalias, o *PyOD 2*⁴ (Chen *et al.*, 2025) é uma biblioteca de código aberto de *Machine Learning* para a linguagem de programação *Python* que auxilia desde a preparação dos dados até a finalização da análise e implementação do modelo. Além disso, também atua como um *wrapper* para diversas bibliotecas e *frameworks* de *Machine Learning*, como *scipy* e *scikit-learn* (Zhao; Nasrullah; Li, 2019). Atualmente, há suporte para o uso da GPU — comumente usada para tarefas de *Machine Learning* — em poucos modelos que são *wrappers* do *TensorFlow*⁵, como o DIF, AE-1SVM e o ALAD, enquanto os demais modelos funcionam apenas na CPU, como consta na documentação da ferramenta⁶.

4.4 Seleção de Hiperparâmetros

A seleção dos hiperparâmetros do método OCSVM foi conduzida por meio de um *Grid Search*, no qual foram exploradas 624 combinações distintas, com os valores usados para a combinatória incluídos na Tabela 1, com o ponto como marcador decimal. A definição de valores possíveis para os hiperparâmetros é realizada com base nas possibilidades permitidas pelo *PyOD*.

³ Disponível em: <https://github.com/pycaret/pycaret>

⁴ Disponível em: <https://github.com/yzhao062/pyod>

⁵ Disponível em: <https://pyod.readthedocs.io/en/latest/faq.html>

⁶ Disponível em: <https://pyod.readthedocs.io/en/latest/pyod.models.html>

Tabela 1 – Conjuntos de valores utilizados para o *Grid Search* do OCSVM

Parâmetro	Valores
<i>Contamination</i>	$\{1 \times 10^{-9}, 0.1, 0.3, 0.49\}$
<i>Kernel</i>	$\{\text{"rbf"}, \text{"linear"}\}$
<i>Gamma</i>	$\{\text{"auto"}, \text{"scale"}, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$
<i>Nu</i>	$\{0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$

Fonte: Elaborada pelo autor (2025).

Para cada combinação de hiperparâmetros, foi desenvolvido um modelo correspondente àquela configuração, treinado exclusivamente com dados normais e avaliado no conjunto de validação. Os parâmetros considerados mais relevantes para o ajuste no modelo foram *contamination*, *gamma*, *kernel* e *nu*, uma vez que possuem forte influência na fronteira de decisão do OCSVM.

Os valores atribuídos ao hiperparâmetro *contamination* foram definidos próximos de zero. No entanto, como o modelo usa apenas dados normais para treinar, o parâmetro deveria ser zero. Conforme a documentação do *PyOD* para o método⁷, o parâmetro *contamination* deve estar no intervalo $0 < contamination \leq 0,5$. Portanto, a biblioteca não permite que o parâmetro seja igual a zero, embora seja possível definir um valor muito próximo dele.

Devido ao alto custo computacional associado à quantidade de combinações de valores avaliadas no *Grid Search* e visando reduzir o tempo de execução da técnica, foi adotada a biblioteca *Joblib*⁸ para paralelizar a CPU em múltiplos processos concorrentes, o que permitiu o treinamento independente para cada modelo.

Os resultados foram ordenados pela FPR e, como critério de corte, definiu-se selecionar, de forma arbitrária, a configuração cuja AUC fosse superior a 60% e a FPR inferior a 1%. Após a ordenação das configurações obtidas, foi selecionada a configuração de maior AUC e com a FPR do critério de corte.

4.5 Avaliação de Desempenho

Abreviada como AUC, a *Area Under the ROC Curve* é uma métrica para avaliar a performance do modelo em um *benchmark* a nível de *frame*, predominantemente empregada em diversos estudos na área de detecção de anomalias (Li; Cheng *et al.*, 2024). O valor da AUC varia de 0 a 1, em que o valor 0,5 representa o desempenho de um classificador aleatório, e o valor máximo corresponde a um classificador perfeito (com uma taxa de erro de classificação menor que zero, por exemplo). Portanto,

⁷ Disponível em: <https://pyod.readthedocs.io/en/latest/pyod.models.html#module-pyod.models.ocsvm>

⁸ Disponível em: <https://joblib.readthedocs.io/en/stable/>

valores mais altos indicam uma maior capacidade de discriminação do método testado (Pawar; Attar, 2022).

Para obter o valor da AUC, faz-se necessário calcular a ROC (*Receiver Operating Characteristic*), que pode ser visualizada como uma curva que representa a taxa de verdadeiros positivos (TPR) em relação à taxa de falsos positivos (FPR), e pode ser matematicamente expressa como uma integral em todos os limiares de decisão (Ok; Klüttermann; Müller, 2024), os quais são limites de corte aplicados aos valores reais resultantes para definir se o dado predito pertence à classe positiva ou negativa (Bohara, 2020).

A TPR, também chamada de *recall* ou sensibilidade, como em Churová *et al.* (2021), mede a habilidade do modelo de identificar todos os verdadeiros positivos — os dados que foram identificados corretamente pelo modelo — em um *dataset*. É definida como a razão entre verdadeiros positivos e o número real de positivos. Em contrapartida, a FPR, também chamada de 1-especificidade (Churová *et al.*, 2021), mensura a proporção de instâncias negativas que foram classificadas incorretamente como positivas e é caracterizada como a razão entre falsos positivos e o número total de instâncias negativas reais (Ok; Klüttermann; Müller, 2024). Finalmente, o cálculo do valor da AUC é realizado ao traçar a curva ROC e calcular a área sob a curva. A matriz de confusão está contida no Quadro 3 para uma melhor compreensão dos conceitos dos dados previstos e reais, em que “positivo” representa a classe anômala e “negativo”, a classe normal.

Quadro 3 – Matriz de confusão

	Real Positivo	Real Negativo
Previsto Positivo	Verdadeiro Positivo (T_P)	Falso Positivo (F_P)
Previsto Negativo	Falso Negativo (F_N)	Verdadeiro Negativo (T_N)

Fonte: Adaptado de Banda, Angryk e Martens (2013).

Além da AUC, FPR e TPR, existem outras métricas comumente empregadas nestas ferramentas para avaliação de modelos de *Machine Learning*, como a precisão e o F1-Score, que é uma média entre a precisão e a TPR (Irizarry, 2020). As definições conceituais e matemáticas de FPR, TPR, Precisão e F1-Score podem ser consultadas no Quadro 4.

Com a finalidade de avaliar o desempenho do modelo, utilizou-se da *Scikit-learn* (Pedregosa *et al.*, 2011) uma biblioteca de *Machine Learning* que disponibiliza diversas métricas para avaliação de desempenho, como a AUC, FPR e F1-Score, que são as principais métricas de avaliação do presente estudo. Ressalta-se que as predições da biblioteca utilizam o limiar padrão de 0,5⁹, o qual também foi mantido nos

⁹ Disponível em: https://scikit-learn.org/stable/modules/classification_thres

Quadro 4 – Métricas de avaliação de modelos de classificação.

Métrica	Fórmula	Definição
<i>Taxa de Falsos Positivos (FPR)</i>	$\frac{F_P}{F_P + T_N}$	Proporção de instâncias negativas que foram incorretamente classificadas como positivas.
<i>Taxa de Verdadeiros Positivos (TPR)</i>	$\frac{T_P}{T_P + F_N}$	Proporção de instâncias positivas que foram corretamente identificadas.
Precisão (P)	$\frac{T_P}{T_P + F_P}$	Proporção de predições positivas que eram de fato positivas.
<i>F1-Score (F₁)</i>	$2 \cdot \frac{P \cdot TPR}{P + TPR}$	Média harmônica entre precisão e TPR.

Fonte: Elaborado pelo autor (2025).

demaís modelos.

O modelo foi treinado a nível de clipe, mas a avaliação foi realizada a nível de *frame*, analogamente a outros trabalhos em detecção de anomalias em vídeo (Pawar; Attar, 2022; Tian *et al.*, 2021; Zaigham Zaheer *et al.*, 2020). O treinamento e teste foram realizados em uma CPU que conta com 6 núcleos, 12 *threads* com um *clock* de 460 MHz em modo turbo. A máquina também dispõe de 16 GB de memória RAM DDR4, com uma frequência de 3200 MHz, e um SSD M.2 NVMe que possui uma velocidade de 3500 MB/s de leitura para carregamento dos arquivos do *dataset*. O sistema operacional utilizado foi *Debian* 14 “Forky”¹⁰, com o *PyOD* 2.0.5 instalado em um ambiente virtual do *Python* 3.10.16.

hold.html#tuning-the-decision-threshold-for-class-prediction

¹⁰ Disponível em: <https://www.debian.org/releases/forky/>

5 RESULTADOS

O *Grid Search* aplicado ao modelo l3D-RGB OCSVM teve duração total de 4 horas, 20 minutos e 30 segundos. A Tabela 2 apresenta um comparativo entre o tempo de treinamento individual, os hiperparâmetros e as métricas de desempenho dos modelos configurados com os parâmetros padrão e com os parâmetros otimizados pela técnica, com destaque para a configuração otimizada.

Tabela 2 – Comparativo entre os modelos OCSVM no conjunto de validação

Configuração	Padrão	Otimizada
Duração	39 s	1 min 9 s
<i>Contamination</i>	1×10^{-9}	1×10^{-9}
<i>Kernel</i>	rbf	rbf
<i>Gamma</i>	auto	auto
<i>Nu</i>	0,50	0,99
AUC	67,45%	69,90%
FPR	0,58%	0,62%
<i>F1-Score</i>	2,02%	2,01%

Fonte: Elaborada pelo autor (2025).

Observa-se que, com a configuração otimizada, a AUC teve um aumento marginal em comparação com a versão padrão, enquanto a FPR apresentou um leve crescimento, o que levou o modelo a classificar um número maior de amostras normais como anômalas em relação ao modelo padrão, como é possível notar com mais clareza nos Quadros 5 e 6. A *F1-Score* é pouco mais de 2,02%, o que indica que o modelo ainda possui uma baixa taxa de acertos de verdadeiros positivos. O custo computacional para o modelo aumentou, mas a diferença em AUC foi pequena demais para justificar que o treinamento durou o dobro do tempo.

Quadro 5 – Matriz de Confusão do OCSVM otimizado

	Real Positivo	Real Negativo
Previsto Positivo	12,11% (31)	87,89% (225)
Previsto Negativo	7,20% (2.803)	92,80% (36.125)

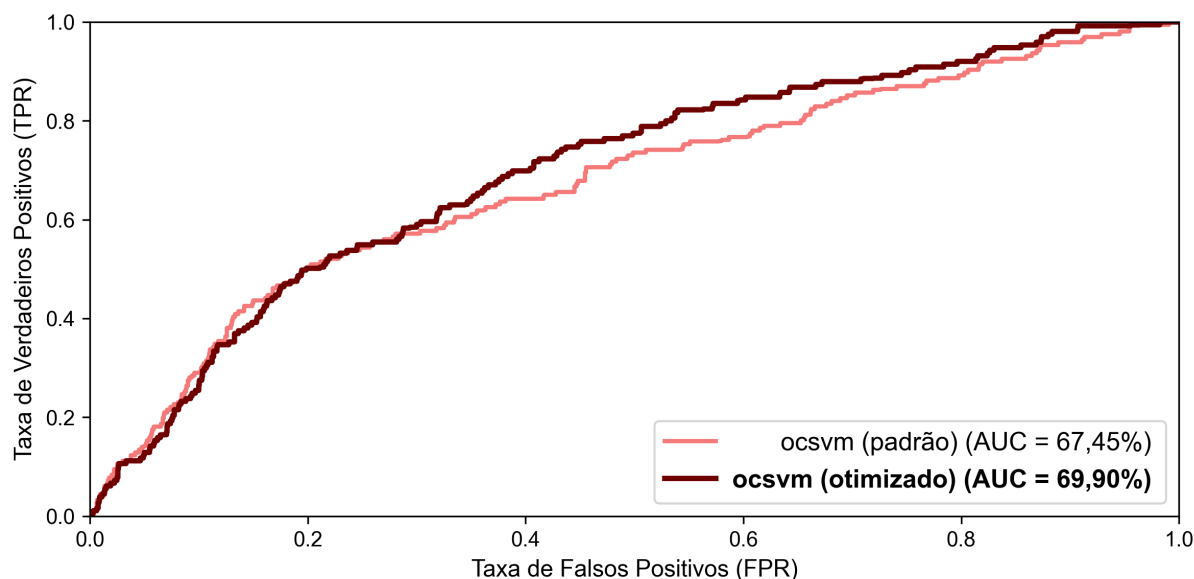
Quadro 6 – Matriz de Confusão do OCSVM padrão

	Real Positivo	Real Negativo
Previsto Positivo	12,92% (31)	87,08% (209)
Previsto Negativo	7,20% (2.803)	92,80% (36.141)

Fonte: Elaborados pelos autor (2025).

Nos hiperparâmetros, a mudança mais substancial ocorreu no parâmetro *nu*, que passou de 0,50 para 0,99, o que implica que 99% dos dados de treino serão formados por vetores de suporte e dados normais, com um impacto também na duração do treino devido à criação de mais vetores de suporte. A Figura 8 compara as curvas ROC entre a parametrização otimizada do OCSVM e a padrão, o que demonstra uma melhora de 2,45% por meio da otimização do *Grid Search* no conjunto de validação.

Figura 8 – Curva ROC dos modelos OCSVM no conjunto de validação



Fonte: Elaborada pelo autor (2025).

Após o *Grid Search*, realizou-se uma comparação do desempenho dos modelos OCSVM, nas versões padrão e otimizada, com outros modelos do estado da arte para detecção de anomalias em vídeos. Os algoritmos possuem naturezas distintas, portanto, é esperado que, por exemplo, métodos supervisionados tenham resultados piores, pois a sua arquitetura não foi feita para aprendizado *One-Class*. Os métodos foram treinados a partir dos mesmos conjuntos de treinamento e teste com atributos I3D.

Os modelos escolhidos para a comparação foram o *Autoencoder*, empregado por Pawar e Attar (2022), o *Variational Autoencoder* (VAE), presente em Kumaran Santhosh et al. (2022), e o *DeepSVDD*, usado por Asad et al. (2022). Além desses, foram considerados dois algoritmos experimentais: o AE-1SVM (Nguyen; Vien, 2019), que combina um AE com o OCSVM, e o AnoGAN (Schlegl et al., 2017), baseado em *Generative Adversarial Networks*.

A realização de um *Grid Search* para a definição dos melhores hiperparâmetros para os outros modelos testados seria extremamente custosa e não compatível com os recursos disponíveis ao considerarmos a complexidade computacional de alguns deles. Dessa forma, os modelos foram implementados a partir do *PyOD*, com o

uso das configurações padrão durante o treinamento, com exceção do hiperparâmetro *contamination* definido como 1×10^{-9} , devido ao uso exclusivo de dados normais para o treinamento. Um comparativo entre a duração do treinamento, AUC, FPR e *F1-Score* dos modelos e os resultados obtidos neste trabalho, no conjunto de teste, é apresentado na Tabela 3, ordenado a partir da AUC e com o modelo investigado em destaque.

Tabela 3 – Desempenho de diferentes métodos OCC no conjunto de teste do *ShanghaiTech*, ordenado a partir da AUC

Modelo	Duração	AUC	FPR	<i>F1-Score</i>
autoencoder	35 min 42 s	80,53%	38,84%	19,62%
ae1svm	1 min 26 s	76,34%	0,79%	10,67%
vae	26 s	75,75%	0,90%	11,45%
deepSVDD (sem AE)	11 s	69,02%	0,21%	1,52%
ocsvm (Otimizado)	1 min 9 s	68,41%	0,33%	1,52%
ocsvm (Padrão)	39 s	68,28%	0,28%	1,15%
anogan	14 min e 11 s	67,55%	0,10%	0,40%
deepSVDD (com AE)	15 s	60,54%	5,91%	14,58%

Fonte: Elaborada pelo autor (2025).

A matriz de confusão do OCSVM otimizado no conjunto de teste é apresentada no Quadro 7. O modelo classificou corretamente 64 cliques como pertencentes à classe positiva, isto é, 64 dos cliques que de fato continham anomalias, e 132.132 cliques como pertencentes à classe negativa, correspondentes a cliques normais.

Todavia, o modelo classificou uma quantidade alta de cliques incorretamente, sendo 7.868 cliques anômalos como normais e 432 cliques normais como anômalos. Com alta taxa de classificações incorretas e a *F1-Score* próxima de zero, é certo afirmar que o modelo possui um bom desempenho na distinção dos dados normais entre os anômalos, mas não é capaz de identificar anomalias.

Quadro 7 – Matriz de confusão para o OCSVM otimizado no conjunto de teste

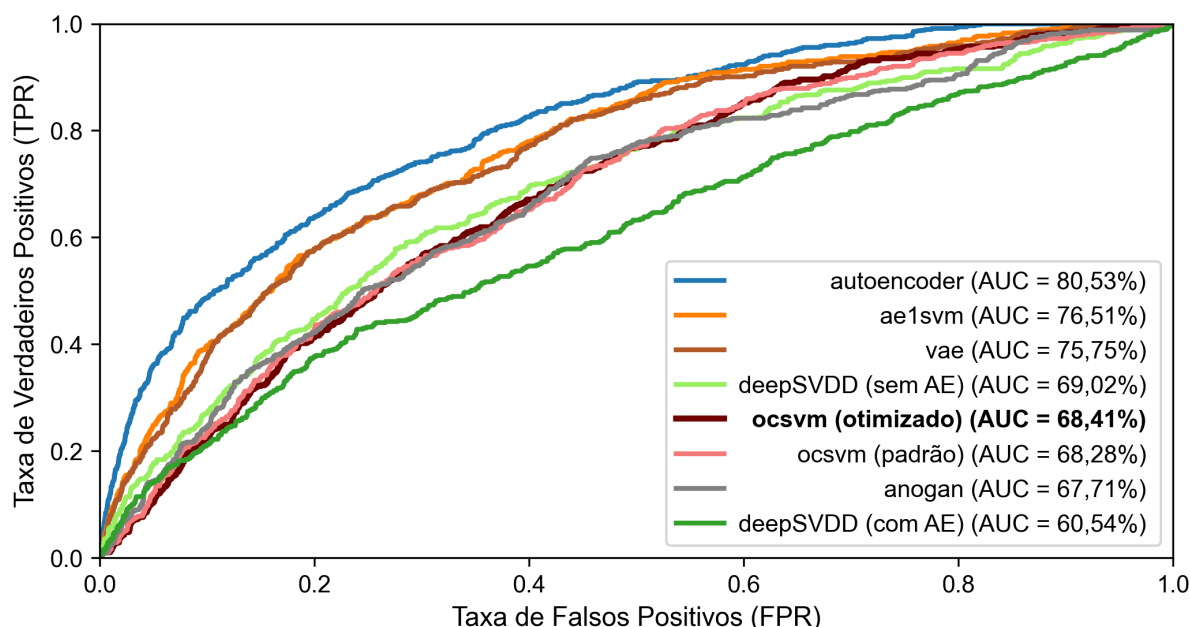
	Real Positivo	Real Negativo
Previsto Positivo	12,90% (64)	87,09% (432)
Previsto Negativo	5,62% (7.868)	94,38% (132.132)

Fonte: Elaborado pelo autor (2025).

De acordo com as curvas ROC dos modelos apresentadas na Figura 9, observa-se que, ainda que otimizado, o OCSVM apresenta desempenho inferior em relação a outros algoritmos do estado da arte quando utilizados com suas configurações padrão. No conjunto de teste, a otimização do método resultou em uma melhoria marginal em comparação à configuração original, de pouco mais de 0,1% em AUC, 0,05% em FPR e 0,37% em *F1-Score*. É necessário frisar que em problemas com dados altamente

desbalanceados, a AUC pode superestimar a qualidade do modelo, sendo necessário uma observação em conjunto com as demais métricas de avaliação de desempenho.

Figura 9 – Curva ROC dos modelos no conjunto de teste



Fonte: Elaborada pelo autor (2025).

De maneira preliminar, podemos notar uma discrepância na duração do treinamento dos modelos, especialmente nos baseados em *Autoencoder*, como mostra a Tabela 3. É possível justificar a partir da parametrização padrão dos modelos¹ no *PyOD*, em que, por exemplo, apesar de o VAE ser uma implementação mais robusta do AE e ter mais épocas de treinamento, ele não possui normalização do *batch*, que diminui a quantidade de cálculos em cada passo de treino, além de possuir configurações que não existem na abordagem original, como número de dimensões latentes ou o coeficiente beta.

O AE-1SVM, apesar de ser composto por um *Autoencoder*, emprega-o unicamente como extrator de características, e usa um SVM para treinar o modelo. O *DeepSVDD* também pode ser usado com AE, embora seu modelo padrão seja construído sem essa integração. Em comparação com o OCSVM otimizado, o *DeepSVDD* possui uma função de otimização que permite treinar o modelo ainda mais rápido.

Somado o uso de 500 épocas de treinamento e da arquitetura complexa de *Generative Adversarial Networks* para detecção de anomalias, o AnoGAN acaba por ser o modelo com a segunda menor pontuação de AUC entre os modelos treinados. Por fim, diferentemente dos demais algoritmos com AE, a versão do *DeepSVDD* com

¹ Disponível em: <https://pyod.readthedocs.io/en/latest/pyod.models.html>

Autoencoder se saiu pior que sua configuração padrão e obteve a menor AUC entre os métodos estudados.

Entre os cinco modelos que melhor desempenharam em AUC, algoritmos baseados em *Autoencoder* encontram-se nas três primeiras posições. De forma contrária ao *One-Class SVM*, o AE não faz uso de uma abordagem baseada em fronteira de decisão; ele comprime as informações de cada *frame* e, posteriormente, aprende as representações (Pawar; Attar, 2022).

Apesar do desempenho elevado em AUC e F1-Score, *Autoencoder* e VAE — métodos que não usam um classificador externo — atingiram valores elevados de FPR com 38,84% e 0,79%, respectivamente. Como a FPR representa a taxa de falsos positivos, no caso do *Autoencoder*, por exemplo, o resultado indica que mais de 35% dos dados de teste foram incorretamente classificados como anômalos. Ainda assim, este estudo ressalta que para uma comparação mais precisa, seria necessário um ajuste nos hiperparâmetros de cada modelo comparado.

Com o propósito de oferecer um embasamento mais consistente aos resultados obtidos, a Tabela 4 apresenta uma comparação ordenada da AUC alcançada pelo método investigado e a de outros métodos do estado da arte, baseados em abordagens não supervisionadas (UVAD), semissupervisionadas (WS-VAD) e *One-Class* (OCC-VAD), que igualmente empregaram o *ShanghaiTech* como *dataset*.

Tabela 4 – Comparativo de estudos de detecção de anomalias em vídeo que empregam o *ShanghaiTech*

Tipo	Estudo	AUC
WS-VAD	Majhi, Das e Bremond (2021)	88,22%
	Zaheer, Mahmood, Astrid <i>et al.</i> (2020)	89,67%
	Purwanto, Chen e Fang (2021)	96,85%
	Tian <i>et al.</i> (2021)	97,21%
UVAD	Li e Jiang (2024)	73,00%
	Tao <i>et al.</i> (2024)	82,12%
	Shi <i>et al.</i> (2025)	88,32%
	Kim <i>et al.</i> (2023)	97,82%
OCC-VAD	Este estudo (OCSVM + I3D-RGB)	68,41%
	Zaigham Zaheer <i>et al.</i> (2020)	69,90%
	Astrid <i>et al.</i> (2021)	75,97%
	Zaheer, Mahmood, Khan <i>et al.</i> (2022)	79,62%
	Yao <i>et al.</i> (2024)	84,30%

Fonte: Elaborada pelo autor (2025).

Em comparação com os estudos da Tabela 4, o presente estudo tem como foco analisar o comportamento do *One-Class SVM*, sem competir diretamente com o estado da arte. Abordagens UVAD e WS-VAD sempre terão desempenho superior, pois utilizam arquiteturas mais complexas. Métodos semissupervisionados, por

exemplo, possuem um certo nível de rotulação dos dados, o que contribui para a capacidade de aprendizado do modelo.

A comparação entre o presente estudo e o estado da arte revela uma AUC similar à de outros trabalhos *One-Class*. O trabalho de [Zaigham Zaheer et al. \(2020\)](#) ultrapassou o estudo atual, com o *framework* proposto *OGNet* a pontuar 69,90% no *ShanghaiTech*, além de atingir 69,47% no *benchmark UCF-Crime*. Cabe destacar que os valores da AUC para os *datasets* não estão disponíveis no artigo original e só são reportados posteriormente em [Zaheer, Mahmood, Khan et al. \(2022\)](#).

Anteriormente discutido, o trabalho de [Astrid et al. \(2021\)](#) propõe um ajuste no método *Autoencoder*, mas ao comparar a AUC obtida para o *ShanghaiTech* no estudo correlato e nos resultados deste estudo para o AE, observam-se valores de 75,97% e 80,53%, respectivamente. Embora a ausência do valor da FPR no trabalho de Astrid impeça uma comparação mais minuciosa, os resultados sugerem que o método original possui desempenho superior, principalmente se houver um ajuste nos hiperparâmetros.

Em [Zaheer, Mahmood, Khan et al. \(2022\)](#), o método atinge uma AUC de 79,62% para o *ShanghaiTech* e 74,20% para o *UCF-Crime*, com um método próprio denominado de *Generative Cooperative Learning*. Por fim, [Yao et al. \(2024\)](#) apresentam um *framework* simples a partir de aprendizado residual de atributos, no qual pontuam 84,30% de AUC no *ShanghaiTech*.

6 CONCLUSÃO

Este trabalho investigou a aprendizagem *One-Class* para detecção de anomalias em vídeo e comparou o desempenho de diferentes algoritmos OCC em um *benchmark* desafiador. Foi experimentada uma técnica robusta do estado da arte para os representar de atributos do *dataset*, e apesar de usar um método clássico, o *One-Class SVM*, sua configuração foi otimizada por meio de um *Grid Search*.

O estudo conclui que o método I3D-RGB OCSVM desempenha melhor na identificação de amostras normais entre anômalas. Entretanto, o modelo não é capaz de identificar anomalias. Ao fim dos resultados, a investigação constatou que a utilização do *One-Class SVM* com atributos profundos no *dataset ShanghaiTech* apresentou resultados medianos tanto nas configurações padrão quanto nas configurações otimizadas, e foi superada pelo *DeepSVDD* e outros métodos baseados em *Autoencoder*.

O direcionamento futuro desta pesquisa contempla quatro ideias principais. A primeira consiste em aprofundar a investigação do OCSVM em contextos mais desafiadores, com a exploração do algoritmo em outros *benchmarks* como UMN, *UCF-Crime* e *UCSD Ped1*. Esse avanço permitiria compreender melhor o potencial e as limitações do método em cenários mais complexos e aumentaria a validade dos resultados obtidos neste estudo.

Uma segunda ideia futura aponta para uma investigação do uso do OCSVM combinado com outras técnicas mais robustas de representação de atributos. Essas estratégias têm o potencial de elevar o desempenho em termos de acurácia e generalização do modelo, com um melhor reconhecimento de anomalias.

Uma terceira linha de investigação foca em *AE* e seus derivados. Mesmo sem otimização, tais algoritmos mostram potencial para detecção de anomalias em vídeo. Com uma configuração adequada, os métodos possuem potencial para superar outras abordagens do estado da arte. De forma mais investigativa, outro caminho seria realizar uma comparação detalhada entre o *One-Class SVM* e *Autoencoder*, com destaque de pontos fortes e limitações de cada método, e como desempenham em outros *datasets*.

Por razões de limitação computacional, não foi realizado *Grid Search* nos hiperparâmetros dos demais modelos na seção de resultados. A realização da busca por melhores configurações poderia fortalecer a validade desta pesquisa ao comparar os demais modelos e proporcionar uma discussão mais aprofundada dos resultados obtidos. Assim, uma linha futura consiste em explorar um *Grid Search* para os outros algoritmos estudados.

Referências

ABDALLA, Moshira *et al.* Video Anomaly Detection in 10 Years: A Survey and Outlook. **Neural Computing and Applications**, v. 37, n. 32, p. 26321–26364, nov. 2025. ISSN 0941-0643, 1433-3058. DOI: [10.1007/s00521-025-11659-8](https://doi.org/10.1007/s00521-025-11659-8).

Disponível em:

<https://link.springer.com/10.1007/s00521-025-11659-8>. Acesso em: 27 nov. 2025.

ALGARNI, Abdulmohsen; AHMAD, Zulfiqar; ALAA ALA'ANZY, Mohammed. An Edge Computing-Based and Threat Behavior-Aware Smart Prioritization Framework for Cybersecurity Intrusion Detection and Prevention of IEDs in Smart Grids With Integration of Modified LGBM and One Class-SVM Models. **IEEE Access**, v. 12, p. 104948–104963, 2024. ISSN 2169-3536. DOI: [10.1109/ACCESS.2024.3435564](https://doi.org/10.1109/ACCESS.2024.3435564). Acesso em: 2 dez. 2025.

AMRAEE, Somaieh *et al.* Abnormal Event Detection in Crowded Scenes Using One-Class SVM. **Signal, Image and Video Processing**, v. 12, n. 6, p. 1115–1123, set. 2018. ISSN 1863-1703, 1863-1711. DOI: [10.1007/s11760-018-1267-z](https://doi.org/10.1007/s11760-018-1267-z). Acesso em: 20 maio 2025.

ASAD, Mujtaba *et al.* Multi-Stream 3D Latent Feature Clustering for Abnormality Detection in Videos. **Applied Intelligence**, v. 52, n. 1, p. 1126–1143, jan. 2022. ISSN 0924-669X, 1573-7497. DOI: [10.1007/s10489-021-02356-9](https://doi.org/10.1007/s10489-021-02356-9). Acesso em: 29 abr. 2025.

ASTRID, Marcella *et al.* Learning Not to Reconstruct Anomalies. *In*: 32ND British Machine Vision Conference 2021, BMVC 2021. Online: BMVA Press, out. 2021. DOI: [10.48550/arXiv.2110.09742](https://arxiv.org/abs/10.48550/arXiv.2110.09742). Disponível em: <https://www.bmvc2021-virtualconference.com/assets/papers/0711.pdf>. Acesso em: 9 out. 2025.

BANDA, J. M.; ANGRYK, R. A.; MARTENS, P. C. H. Steps Toward a Large-Scale Solar Image Data Analysis to Differentiate Solar Phenomena. *en*. **Solar Physics**, v. 288, n. 1, p. 435–462, nov. 2013. ISSN 1573-093X. DOI: [10.1007/s11207-013-0304-x](https://doi.org/10.1007/s11207-013-0304-x). Disponível em: <https://doi.org/10.1007/s11207-013-0304-x>. Acesso em: 21 out. 2025.

BHATNAGAR, Bharat Lal *et al.* BEHAVE: Dataset and Method for Tracking Human Object Interactions. *In*: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, jun. 2022. p. 15914–15925. ISBN 978-1-6654-6946-3. DOI: [10.1109/CVPR52688.2022.01547](https://doi.org/10.1109/CVPR52688.2022.01547). Acesso em: 1 dez. 2025.

BISCHL, Bernd *et al.* Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges. **WIREs Data Mining and Knowledge Discovery**,

v. 13, n. 2, e1484, mar. 2023. ISSN 1942-4787, 1942-4795. DOI: [10.1002/widm.1484](https://doi.org/10.1002/widm.1484). Acesso em: 27 nov. 2025.

BOHARA, Bharat. Adaptive Threshold for Better Performance of the Recognition and Re-identification Models. **CoRR**, abs/2012.14305, 2020. DOI: [10.48550/arXiv.2012.14305](https://doi.org/10.48550/arXiv.2012.14305). arXiv: [2012.14305](https://arxiv.org/abs/2012.14305).

CARREIRA, Joao; ZISSERMAN, Andrew. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *In*: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, jul. 2017. p. 4724–4733. ISBN 978-1-5386-0457-1. DOI: [10.1109/CVPR.2017.502](https://doi.org/10.1109/CVPR.2017.502). Acesso em: 13 maio 2025.

CHEN, Sihan *et al.* PyOD 2: A Python Library for Outlier Detection with LLM-powered Model Selection. *In*: COMPANION Proceedings of the ACM on Web Conference 2025. Sydney NSW Australia: ACM, maio 2025. p. 2807–2810. ISBN 979-8-4007-1331-6. DOI: [10.1145/3701716.3715196](https://doi.org/10.1145/3701716.3715196). Acesso em: 1 dez. 2025.

CHUROVÁ, Vendula *et al.* Anomaly Detection Algorithm for Real-World Data and Evidence in Clinical Research: Implementation, Evaluation, and Validation Study. **JMIR Medical Informatics**, v. 9, n. 5, e27172, maio 2021. ISSN 2291-9694. DOI: [10.2196/27172](https://doi.org/10.2196/27172). Acesso em: 17 out. 2025.

DEEPAK, K. *et al.* Deep Multi-view Representation Learning for Video Anomaly Detection Using Spatiotemporal Autoencoders. **Circuits, Systems, and Signal Processing**, v. 40, n. 3, p. 1333–1349, mar. 2021. ISSN 0278-081X, 1531-5878. DOI: [10.1007/s00034-020-01522-7](https://doi.org/10.1007/s00034-020-01522-7). Acesso em: 29 abr. 2025.

ERFANIAN EBADI, Salehe *et al.* Automated Detection of Pneumonia in Lung Ultrasound Using Deep Video Classification for COVID-19. **Informatics in Medicine Unlocked**, v. 25, p. 100687, 2021. ISSN 23529148. DOI: [10.1016/j.imu.2021.100687](https://doi.org/10.1016/j.imu.2021.100687). Acesso em: 10 out. 2025.

FERRYMAN, J.; SHAHROKNI, A. PETS2009: Dataset and Challenge. *In*: 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. Snowbird, UT, USA: IEEE, dez. 2009. p. 1–6. ISBN 978-1-4244-5503-4. DOI: [10.1109/PETS-WINTER.2009.5399556](https://doi.org/10.1109/PETS-WINTER.2009.5399556). Acesso em: 27 nov. 2025.

FUJITA, Hirofumi; MATSUKAWA, Tetsu; SUZUKI, Einoshin. Detecting Outliers with One-Class Selective Transfer Machine. **Knowledge and Information Systems**, v. 62, n. 5, p. 1781–1818, maio 2020. ISSN 0219-1377, 0219-3116. DOI: [10.1007/s10115-019-01407-5](https://doi.org/10.1007/s10115-019-01407-5). Acesso em: 1 out. 2025.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge, Massachusetts: The MIT Press, 2016. (Adaptive Computation and Machine Learning). ISBN 978-0-262-03561-3.

- HAN, Jiawei; KAMBER, Micheline; PEI, Jian. **Data Mining: Concepts and Techniques**. 3rd ed. Amsterdam Boston: Elsevier/Morgan Kaufmann, 2012. (Morgan Kaufmann Series in Data Management Systems). ISBN 978-0-12-381479-1.
- HE, Kaiming *et al.* Deep Residual Learning for Image Recognition. *In*: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, jun. 2016. p. 770–778. ISBN 978-1-4673-8851-1. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90). Acesso em: 1 dez. 2025.
- HOFMANN, Thomas; SCHÖLKOPF, Bernhard; SMOLA, Alexander J. Kernel Methods in Machine Learning. **The Annals of Statistics**, v. 36, n. 3, jun. 2008. ISSN 0090-5364. DOI: [10.1214/009053607000000677](https://doi.org/10.1214/009053607000000677). Acesso em: 27 nov. 2025.
- ILYAS, Zirgham *et al.* A Hybrid Deep Network Based Approach for Crowd Anomaly Detection. **Multimedia Tools and Applications**, v. 80, n. 16, p. 24053–24067, jul. 2021. ISSN 1380-7501, 1573-7721. DOI: [10.1007/s11042-021-10785-4](https://doi.org/10.1007/s11042-021-10785-4). Acesso em: 16 abr. 2025.
- IRIZARRY, Rafael A. **Introduction to Data Science: Data Analysis and Prediction Algorithms with R**. Boca Raton London New York: CRC Press, Taylor & Francis, 2020. (Data Science Series). ISBN 978-0-367-35798-6 978-1-000-70773-1.
- JACKSON, Chris. Global Advisor: Crime and Law Enforcement. Ipsos, jun. 2023. Disponível em: <https://www.ipsos.com/sites/default/files/ct/news/documents/2023-06/Ipsos-Global-Advisor-Crime-Law-Enforcement-Full-Report.pdf>. Acesso em: 25 abr. 2025.
- JUNG, Yeondae; WHEELER, Andrew P. The Effect of Public Surveillance Cameras on Crime Clearance Rates. **Journal of Experimental Criminology**, v. 19, n. 1, p. 143–164, mar. 2023. ISSN 1573-3750, 1572-8315. DOI: [10.1007/s11292-021-09477-8](https://doi.org/10.1007/s11292-021-09477-8). Acesso em: 25 abr. 2025.
- KAISLER, Stephen *et al.* Big Data and Analytics: Issues and Challenges for the Past and Next Ten Years. *In*: HAWAII International Conference on System Sciences. [S. l.: s. n.], 2023. DOI: [10.24251/HICSS.2023.101](https://doi.org/10.24251/HICSS.2023.101). HDL: [10125/102730](https://hdl.handle.net/10125/102730). Acesso em: 22 out. 2025.
- KAY, Will *et al.* The Kinetics Human Action Video Dataset. arXiv, maio 2017. DOI: [10.48550/arXiv.1705.06950](https://doi.org/10.48550/arXiv.1705.06950). arXiv: [1705.06950](https://arxiv.org/abs/1705.06950) [cs]. Acesso em: 27 nov. 2025.
- KHAN, Shehroz S.; MADDEN, Michael G. One-Class Classification: Taxonomy of Study and Review of Techniques. **The Knowledge Engineering Review**, v. 29, n. 3, p. 345–374, jun. 2014. ISSN 0269-8889, 1469-8005. DOI: [10.1017/S026988891300043X](https://doi.org/10.1017/S026988891300043X). Acesso em: 27 nov. 2025.

KIM, Jaehyun *et al.* Unsupervised Video Anomaly Detection Based on Similarity with Predefined Text Descriptions. **Sensors**, v. 23, n. 14, p. 6256, jul. 2023. ISSN 1424-8220. DOI: [10.3390/s23146256](https://doi.org/10.3390/s23146256). Acesso em: 3 out. 2025.

KITTIDACHANAN, Kittikun *et al.* Anomaly Detection Based on GS-OCSVM Classification. *In*: 2020 12th International Conference on Knowledge and Smart Technology (KST). Pattaya, Chonburi, Thailand: IEEE, jan. 2020. p. 64–69. ISBN 978-1-7281-4466-5. DOI: [10.1109/KST48564.2020.9059326](https://doi.org/10.1109/KST48564.2020.9059326). Acesso em: 27 nov. 2025.

KUEHNE, H. *et al.* HMDB: A Large Video Database for Human Motion Recognition. *In*: 2011 International Conference on Computer Vision. Barcelona, Spain: IEEE, nov. 2011. p. 2556–2563. ISBN 978-1-4577-1102-2 978-1-4577-1101-5 978-1-4577-1100-8. DOI: [10.1109/ICCV.2011.6126543](https://doi.org/10.1109/ICCV.2011.6126543). Acesso em: 27 nov. 2025.

KUMARAN SANTHOSH, Kelathodi *et al.* Vehicular Trajectory Classification and Traffic Anomaly Detection in Videos Using a Hybrid CNN-VAE Architecture. **IEEE Transactions on Intelligent Transportation Systems**, v. 23, n. 8, p. 11891–11902, ago. 2022. ISSN 1524-9050, 1558-0016. DOI: [10.1109/TITS.2021.3108504](https://doi.org/10.1109/TITS.2021.3108504). Acesso em: 17 set. 2025.

LEAL, Adriana *et al.* Unsupervised EEG Preictal Interval Identification in Patients with Drug-Resistant Epilepsy. **Scientific Reports**, v. 13, n. 1, p. 784, jan. 2023. ISSN 2045-2322. DOI: [10.1038/s41598-022-23902-6](https://doi.org/10.1038/s41598-022-23902-6). Acesso em: 28 out. 2025.

LEYVA, Roberto; SANCHEZ, Victor; LI, Chang-Tsun. The LV Dataset: A Realistic Surveillance Video Dataset for Abnormal Event Detection. *In*: 2017 5th International Workshop on Biometrics and Forensics (IWBF). Coventry, United Kingdom: IEEE, abr. 2017. p. 1–6. ISBN 978-1-5090-5791-7. DOI: [10.1109/IWBF.2017.7935096](https://doi.org/10.1109/IWBF.2017.7935096). Acesso em: 27 nov. 2025.

LI, Changkang; JIANG, Yalong. VLAVID: Vision-Language Models Assisted Unsupervised Video Anomaly Detection. *In*: 35TH British Machine Vision Conference, BMVC 2024. Glasgow, UK: BMVA Press, 2024. DOI: [10.48550/arXiv.2409.14109](https://doi.org/10.48550/arXiv.2409.14109). Disponível em: <https://bmvc2024.org/proceedings/599/>.

LI, Shifeng; CHENG, Yan *et al.* Video Anomaly Detection Based on a Multi-Layer Reconstruction Autoencoder with a Variance Attention Strategy. **Image and Vision Computing**, v. 146, p. 105011, jun. 2024. ISSN 02628856. DOI: [10.1016/j.imavis.2024.105011](https://doi.org/10.1016/j.imavis.2024.105011). Acesso em: 17 set. 2025.

LIASHCHYNSKYI, Petro; LIASHCHYNSKYI, Pavlo. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. arXiv, dez. 2019. DOI: [10.48550/arXiv.1912.06059](https://doi.org/10.48550/arXiv.1912.06059). arXiv: [1912.06059 \[cs\]](https://arxiv.org/abs/1912.06059). Acesso em: 18 out. 2025.

LU, Cewu; SHI, Jianping; JIA, Jiaya. Abnormal Event Detection at 150 FPS in MATLAB. *In: 2013 IEEE International Conference on Computer Vision*. Sydney, Australia: IEEE, dez. 2013. p. 2720–2727. ISBN 978-1-4799-2840-8. DOI: [10.1109/ICCV.2013.338](https://doi.org/10.1109/ICCV.2013.338). Acesso em: 27 nov. 2025.

LUO, Weixin; LIU, Wen; GAO, Shenghua. A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework. *In: 2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, out. 2017. p. 341–349. ISBN 978-1-5386-1032-9. DOI: [10.1109/ICCV.2017.45](https://doi.org/10.1109/ICCV.2017.45). Acesso em: 27 nov. 2025.

MAJHI, Snehashis; DAS, Srijan; BREMOND, Francois. DAM: Dissimilarity Attention Module for Weakly-supervised Video Anomaly Detection. *In: 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Washington, DC, USA: IEEE, nov. 2021. p. 1–8. ISBN 978-1-6654-3396-9. DOI: [10.1109/AVSS52988.2021.9663810](https://doi.org/10.1109/AVSS52988.2021.9663810). Acesso em: 27 nov. 2025.

MASAKUNA, Jordan F. *et al.* Impact of Inaccurate Contamination Ratios on Robust Unsupervised Anomaly Detection: Experimental Investigation. Preprints, jan. 2025. DOI: [10.36227/techrxiv.173396113.31607552/v2](https://doi.org/10.36227/techrxiv.173396113.31607552/v2). Acesso em: 27 nov. 2025.

MCKINNEY, Wes. Data Structures for Statistical Computing in Python. *In: PYTHON in Science Conference*. Austin, Texas: [s. n.], 2010. p. 56–61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a). Acesso em: 27 nov. 2025.

MITCHELL, Tom M. **Machine Learning**. Nachdr. New York: McGraw-Hill, 2013. (McGraw-Hill Series in Computer Science). ISBN 978-0-07-042807-2 978-0-07-115467-3.

NAYAK, Rikin; CHAUDHARI, Jitendra P. Real-World Anomaly Detection in Video Using Spatio-Temporal Features Analysis for Weakly Labelled Data with Auto Label Generation. **International Journal of Electrical and Computer Engineering Systems**, v. 14, n. 5, p. 565–573, jun. 2023. DOI: [10.32985/ijeces.14.5.8](https://doi.org/10.32985/ijeces.14.5.8). Disponível em: <https://ijeces.ferit.hr/index.php/ijeces/article/view/2186>.

NGUYEN, Minh-Nghia; VIEN, Ngo Anh. Scalable and Interpretable One-Class SVMs with Deep Learning and Random Fourier Features. *In: BERLINGERIO, Michele et al. (ed.). Machine Learning and Knowledge Discovery in Databases*. Cham: Springer International Publishing, 2019. v. 11051. p. 157–172. ISBN 978-3-030-10924-0 978-3-030-10925-7. DOI: [10.1007/978-3-030-10925-7_10](https://doi.org/10.1007/978-3-030-10925-7_10). Acesso em: 17 set. 2025.

NOBLE, William S. What Is a Support Vector Machine? **Nature Biotechnology**, v. 24, n. 12, p. 1565–1567, dez. 2006. ISSN 1087-0156, 1546-1696. DOI: [10.1038/nbt1206-1565](https://doi.org/10.1038/nbt1206-1565). Acesso em: 27 nov. 2025.

OK, Minjae; KLÜTTERMANN, Simon; MÜLLER, Emmanuel. Exploring the Impact of Outlier Variability on Anomaly Detection Evaluation Metrics. *arXiv*, 2024. DOI: [10.48550/ARXIV.2409.15986](https://doi.org/10.48550/ARXIV.2409.15986). Acesso em: 27 nov. 2025.

PAWAR, Karishma; ATTAR, Vahida. Deep Learning Model Based on Cascaded Autoencoders and One-class Learning for Detection and Localization of Anomalies from Surveillance Videos. **IET Biometrics**, v. 11, n. 4, p. 289–303, jul. 2022. ISSN 2047-4938, 2047-4946. DOI: [10.1049/bme2.12064](https://doi.org/10.1049/bme2.12064). Acesso em: 16 abr. 2025.

PEDREGOSA, Fabian *et al.* Scikit-learn: Machine Learning in Python. **J. Mach. Learn. Res.**, JMLR.org, v. 12, null, p. 2825–2830, nov. 2011. ISSN 1532-4435. DOI: <https://doi.org/10.48550/arXiv.1201.0490>. Acesso em: 5 out. 2025.

PURWANTO, Didik; CHEN, Yie-Tarng; FANG, Wen-Hsien. Dance with Self-Attention: A New Look of Conditional Random Fields on Anomaly Detection in Videos. *In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, out. 2021. p. 173–183. ISBN 978-1-6654-2812-5. DOI: [10.1109/ICCV48922.2021.00024](https://doi.org/10.1109/ICCV48922.2021.00024). Acesso em: 27 nov. 2025.

ROITBERG, Alina; AL-HALAH, Ziad; STIEFELHAGEN, Rainer. Informed Democracy: Voting-based Novelty Detection for Action Recognition. Karlsruhe, Germany, 2018. *arXiv: 1810.12819 [cs.CV]*. Disponível em: <https://arxiv.org/abs/1810.12819>.

RUFF, Lukas *et al.* Deep One-Class Classification. *In: DY, Jennifer; KRAUSE, Andreas (ed.). Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden: PMLR, out. 2018. v. 80. (Proceedings of Machine Learning Research), p. 4393–4402. Disponível em: <https://proceedings.mlr.press/v80/ruff18a.html>.

SAMAILA, Yau Alhaji *et al.* Video Anomaly Detection: A Systematic Review of Issues and Prospects. **Neurocomputing**, v. 591, p. 127726, jul. 2024. ISSN 09252312. DOI: [10.1016/j.neucom.2024.127726](https://doi.org/10.1016/j.neucom.2024.127726). Acesso em: 15 abr. 2025.

SCHLEGL, Thomas *et al.* Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. *In: NIETHAMMER, Marc *et al.* (ed.). Information Processing in Medical Imaging*. Cham: Springer International Publishing, 2017. v. 10265. p. 146–157. ISBN 978-3-319-59049-3 978-3-319-59050-9. DOI: [10.1007/978-3-319-59050-9_12](https://doi.org/10.1007/978-3-319-59050-9_12). Acesso em: 17 set. 2025.

SCHÖLKOPF, Bernhard *et al.* Estimating the Support of a High-Dimensional Distribution. **Neural Computation**, v. 13, n. 7, p. 1443–1471, jul. 2001. ISSN 0899-7667, 1530-888X. DOI: [10.1162/089976601750264965](https://doi.org/10.1162/089976601750264965). Acesso em: 27 nov. 2025.

SELIYA, Naeem; ABDOLLAH ZADEH, Azadeh; KHOSHGOFTAAR, Taghi M. A Literature Review on One-Class Classification and Its Potential Applications in Big Data. **Journal of Big Data**, v. 8, n. 1, p. 122, set. 2021. ISSN 2196-1115. DOI: [10.1186/s40537-021-00514-x](https://doi.org/10.1186/s40537-021-00514-x). Acesso em: 14 abr. 2025.

SHAMS, Mahmoud Y. *et al.* Water Quality Prediction Using Machine Learning Models Based on Grid Search Method. **Multimedia Tools and Applications**, v. 83, n. 12, p. 35307–35334, set. 2023. ISSN 1573-7721. DOI: [10.1007/s11042-023-16737-4](https://doi.org/10.1007/s11042-023-16737-4). Acesso em: 1 set. 2025.

SHI, Haoyue *et al.* Learning Anomalies with Normality Prior for Unsupervised Video Anomaly Detection. In: LEONARDIS, Aleš *et al.* (ed.). **Computer Vision – ECCV 2024**. Cham: Springer Nature Switzerland, 2025. v. 15064. p. 163–180. ISBN 978-3-031-72657-6 978-3-031-72658-3. DOI: [10.1007/978-3-031-72658-3_10](https://doi.org/10.1007/978-3-031-72658-3_10). Acesso em: 3 out. 2025.

SOOMRO, Khurram; ZAMIR, Amir Roshan; SHAH, Mubarak. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. **CoRR**, abs/1212.0402, dez. 2012. DOI: [10.48550/arXiv.1212.0402](https://doi.org/10.48550/arXiv.1212.0402). arXiv: [1212.0402 \[cs\]](https://arxiv.org/abs/1212.0402). Acesso em: 27 nov. 2025.

TAO, Chenchen *et al.* Feature Reconstruction With Disruption for Unsupervised Video Anomaly Detection. **IEEE Transactions on Multimedia**, v. 26, p. 10160–10173, 2024. ISSN 1520-9210, 1941-0077. DOI: [10.1109/TMM.2024.3405716](https://doi.org/10.1109/TMM.2024.3405716). Acesso em: 27 nov. 2025.

THOMAS, Albert; FEUILLARD, Vincent; GRAMFORT, Alexandre. Calibration of One-Class SVM for MV Set Estimation. In: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA). Campus des Cordeliers, Paris, France: IEEE, out. 2015. p. 1–9. ISBN 978-1-4673-8272-4. DOI: [10.1109/DSAA.2015.7344789](https://doi.org/10.1109/DSAA.2015.7344789). Acesso em: 1 dez. 2025.

TIAN, Yu *et al.* Weakly-Supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, out. 2021. p. 4955–4966. ISBN 978-1-6654-2812-5. DOI: [10.1109/ICCV48922.2021.00493](https://doi.org/10.1109/ICCV48922.2021.00493). Acesso em: 27 nov. 2025.

WAN, Boyang *et al.* Anomaly Detection in Video Sequences: A Benchmark and Computational Model. **IET Image Processing**, v. 15, n. 14, p. 3454–3465, dez. 2021. ISSN 1751-9659, 1751-9667. DOI: [10.1049/ipr2.12258](https://doi.org/10.1049/ipr2.12258). Acesso em: 27 nov. 2025.

XIA, Xiangli; GAO, Yang. Video Abnormal Event Detection Based on One-Class Neural Network. Edição: Bai Yuan Ding. **Computational Intelligence and Neuroscience**, v. 2021, n. 1, p. 1955116, jan. 2021. ISSN 1687-5265, 1687-5273. DOI: [10.1155/2021/1955116](https://doi.org/10.1155/2021/1955116). Acesso em: 29 abr. 2025.

YANG, Kun; KPOTUFE, Samory; FEAMSTER, Nick. An Efficient One-Class SVM for Anomaly Detection in the Internet of Things. arXiv, abr. 2021. DOI: [10.48550/arXiv.2104.11146](https://doi.org/10.48550/arXiv.2104.11146). arXiv: 2104.11146 [cs]. Acesso em: 27 nov. 2025.

YAO, Xincheng *et al.* ResAD: a simple framework for class generalizable anomaly detection. *In: PROCEEDINGS of the 38th International Conference on Neural Information Processing Systems*. Vancouver, BC, Canada: Curran Associates Inc., 2024. (NIPS '24). ISBN 9798331314385. DOI: <https://doi.org/10.52202/079017-3979>.

ZAHEER, Zaigham; MAHMOOD, Arif; ASTRID, Marcella *et al.* CLAWS: Clustering Assisted Weakly Supervised Learning with Normalcy Suppression for Anomalous Event Detection. *In: VEDALDI, Andrea et al. (ed.). Computer Vision – ECCV 2020*. Cham: Springer International Publishing, 2020. v. 12367. p. 358–376. ISBN 978-3-030-58541-9 978-3-030-58542-6. DOI: [10.1007/978-3-030-58542-6_22](https://doi.org/10.1007/978-3-030-58542-6_22). Acesso em: 27 nov. 2025.

ZAHEER, Zaigham; MAHMOOD, Arif; KHAN, M. Haris *et al.* Generative Cooperative Learning for Unsupervised Video Anomaly Detection. *In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, jun. 2022. p. 14724–14734. ISBN 978-1-6654-6946-3. DOI: [10.1109/CVPR52688.2022.01433](https://doi.org/10.1109/CVPR52688.2022.01433). Acesso em: 3 out. 2025.

ZAIGHAM ZAHEER, Muhammad *et al.* Old Is Gold: Redefining the Adversarially Learned One-Class Classifier Training Paradigm. *In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, jun. 2020. p. 14171–14181. ISBN 978-1-7281-7168-5. DOI: [10.1109/CVPR42600.2020.01419](https://doi.org/10.1109/CVPR42600.2020.01419). Acesso em: 27 nov. 2025.

ZHAO, Yue; NASRULLAH, Zain; LI, Zheng. PyOD: A Python Toolbox for Scalable Outlier Detection. **Journal of Machine Learning Research**, v. 20, n. 96, p. 1–7, 2019. Disponível em: <http://jmlr.org/papers/v20/19-011.html>. Acesso em: 17 set. 2025.

ZHENG, Alice; CASARI, Amanda. **Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists**. First edition. Beijing : Boston: O'Reilly, 2018. ISBN 978-1-4919-5324-2.

ZHU, Sijie; CHEN, Chen; SULTANI, Waqas. Video Anomaly Detection for Smart Surveillance. *In: COMPUTER Vision*. Cham: Springer International Publishing, 2020. p. 1–8. ISBN 978-3-030-03243-2. DOI: [10.1007/978-3-030-03243-2_845-1](https://doi.org/10.1007/978-3-030-03243-2_845-1). Acesso em: 27 nov. 2025.