Visualização de features de ataques DoS baseada em Gramian Angular Field

Gabriel Zancheta Scavazini
Instituto de Biociências, Letras e Ciências Exatas
UNESP - Universidade Estadual Paulista
São José do Rio Preto, Brasil
gabriel.scavazini@unesp.br

Wellington Reguera Gouveia
Instituto de Biociências, Letras e Ciências Exatas
UNESP - Universidade Estadual Paulista
São José do Rio Preto, Brasil
wellington.reguera@unesp.br

***Abstract*—Ataque de negativa de serviço é um método de inutilizar um sistema computacional por meio de consecutivas requisições em um serviço e, por mais antigo que seja, continua sendo um problema constante em sistemas atuais. Os dados de tráfego de uma rede são considerados séries temporais. E portanto, ao sofrer um ataque *DoS* ocorre uma variação anormal de tal série, o que possibilita a aplicação de técnicas de visualização de *features* para a detecção em seu estágio inicial. Neste sentido, técnicas de transformação de séries temporais em imagens são constantemente utilizadas para visualizar as características desses dados e possibilitar que métodos de *deep learning* sejam aplicados para a classificação dessas imagens. Durante este trabalho, foi utilizado *Gramian Angular Field* para visualizar ataques de *DDoS* (negativa de serviço distribuído) em imagens. Em seguida, foi utilizada uma Rede Neural Convolucional pré-treinada para a classificar essas imagens. A partir dos resultados obtidos, o modelo desenvolvido neste trabalho gerou 98% de acurácia para a detecção de ataques.**

***Index Terms*—Gramiam Angular Field, Transformação de imagens, Séries Temporais, Ataques Cibernéticos**

I. Introdução

Ataques cibernéticos são ações não autorizadas contra uma rede ou computador que resultam em uma violação da política de segurança, atualmente o Brasil se encontra em 2º lugar no maior número de alvos da América Latina [3]. As ameaças de rede dependem da manipulação dos fluxos de pacotes enviados através de uma rede, as formas mais comuns são negação de serviço (*DoS*) e negação de serviço distribuído (*DDoS*) [9].

O Ataque Distribuído de Negação de Serviço perturba o funcionamento de uma rede ao inundar o servidor ou site com múltiplas solicitações simultâneas e em locais diferentes, o que faz com que a largura de banda dos usuários legítimos seja reduzida. Ataques *DDoS* são bastante problemáticos para serem detectados, pois usam locais distintos, além de ser possível a utilização de vários dispositivos (ao mesmo tempo) para a realização do ataque [1].

Para criar uma técnica de detecção de ataques em um ambiente dinâmico de redes, métodos de *deep learning* são muito utilizados [1]. Neste contexto, a aplicação de métodos de *deep learning* para a criação de uma técnica que possibilite a extração e a visualização de atributos, tornam necessário a transformação dos dados de ataques em imagens.

Uma Rede neural Convolucional (*CNN*) é uma arquitetura amplamente utilizada que demonstra desempenho eficaz em visão computacional com a vantagem de representações equivalentes, compartilhamento de parâmetros e interações esparsas [9]. A *CNN* é capaz de extrair características de dados com estruturas de convolução e, diferente dos métodos tradicionais de extração de características, permite não extrair manualmente cada uma das características [5].

Uma *CNN* é composta de três tipos principais de camadas, as camadas convolucionais, as camadas de *pooling* e as camadas totalmente conectadas [10]. As camadas convolucionais servem como extratores de atributos que utilizam vários *kernels* para aprender as representações de recursos das imagens de entrada. Por sua vez as camadas de *pooling* são responsáveis por reduzir as dimensões espaciais dos mapas de recursos gerados pelas camadas convolucionais. As camadas totalmente conectadas interpretam os mapas de recursos e desempenham a função de raciocínio de alto nível [9].

Comparada com outros tipos de redes, a *CNN* possui diversas vantagens:

1) Conexões locais - cada neurônio não está conectado a todos os neurônios da camada anterior, mas apenas a um pequeno número de neurônios, o que torna eficaz a redução de parâmetros e a aceleração da convergência [5].
2) Compartilhamento de peso - um grupo de conexões pode compartilhar os mesmos pesos, o que reduz ainda mais os parâmetros [5].
3) Redução resolução - uma camada de *pooling* aproveita o princípio da correlação local da imagem para reduzir a sua resolução, o que pode reduzir a quantidade de dados enquanto retém informações úteis [5].

Neste trabalho, foi utilizado *Gramian Angular Field* para a transformação de dados de séries temporais em imagens e em seguida foi aplicada uma Rede Neural Convolucional para extração de *features*. A partir dos resultados, foi possível analisar que o método desenvolvido se mostrou eficiente para a detecção de ataques *DoS*.

II. Trabalhos Correlatos

Um conjunto de artigos sobre o tema em questão foi pesquisado e estudado dentro dos principais meios de divulgação científica, estes artigos têm em comum a

implementação de recursos de transformação de séries temporais em imagens, Redes Neurais Convolucionais e detecção de invasões em redes.

Xia et al. apresenta uma técnica de classificação de *softwares* maliciosos por meio de imagens obtidas por dados de séries temporais. A quantidade de *softwares* maliciosos (*Malware*) estão crescendo exponencialmente e se torna necessária a identificação de *softwares* normais e *softwares* maliciosos. Arquivos de *malwares* podem ser convertidos em sinais e é possível extrair um vetor de *features* destes sinais. A partir destes sinais, é aplicada a técnica de *Markov Transition Field* (*MTF*) para transformar em imagens. Assim, as imagens são convertidas em um vetor unidimensional para que uma *Suport Vector Machine* classificar o método [12].

Mao et al. propõe um método para detecção de anomalias em dados da área da saúde. Detecção de danos é uma das tarefas mais importantes para monitoramento da saúde, as falhas nos sistemas de sensores, que podem gerar anomalias nos dados, é um sério problema e identificar essas anomalias é essencial para garantir a confiabilidade dos sistemas de saúde. Neste contexto, métodos de Aprendizado de Máquina têm potencial para automatizar o processo da detecção de anomalias. Neste trabalho, foram utilizadas Redes Adversárias Generativas combinadas com um método não supervisionado, chamados *autoencoders*, para melhorar o desempenho dos métodos de aprendizagem não supervisionados existentes. Além disso, foi aplicado *GAF* para a transformação dos dados de séries temporais em imagens. Os resultados mostraram que a metodologia proposta pode identificar com sucesso as anomalias dos dados com uma boa acurácia e robustez [6].

Hussain et al. apresenta uma técnica para detecção de ataques *DoS* e *DDoS* em dispositivos *IoT*. Ataques de redes estão crescendo tanto na sua frequência quanto na sua intensidade em dispositivos que aplicam o conceito de Internet das coisas (*IoT*). As soluções tradicionais de segurança como *firewalls*, sistemas de detecção de invasões, entre outros são incapazes de detectar ataques de *DoS* e *DDoS* complexos, uma vez que eles filtram o tráfego normal e de ataque com regras estáticas predefinidas. Neste trabalho, foi utilizado o potencial das *CNNs* para conseguir detectar estes ataques complexos, baseando na conversão dos dados em imagens. Para a classificação do modelo, foi utilizada a rede *ResNet*. O método proposto gerou ótimos resultados para a classificação binária destes ataques. Além de conseguir uma boa precisão para diferenciar até 11 tipos de ataques *DoS* e *DDoS* [4].

Terzi propõe uma perspectiva diferente para detectar invasões. Foi adaptado *Gramian Angular Difference Field* para transformações dos dados em imagens. As imagens geradas são classificadas de maneira binária e multiclasse. Além de conseguir detectar o tráfego normal de ataques, é possível detectar 5 tipos de ataques, entre eles: *BotNet*, *DDoS*, *DoS*, *FTP-Patator*, *Port Scan*, *SSH-Patator* e ataques Web. O trabalho proposto conseguiu gerar resultados satisfatórios sem a necessidade de mecanismos muito complexos [9].

Por meio da análise dos trabalhos correlatos foi possível propor a Tabela I, no qual conteúdo apresenta os elementos que permitem a comparação entre os principais conceitos apresentados.

III. Metodologia

O método desenvolvido neste estudo são descritos nas seguintes sessões: (A) Processamento dos dados; (B) Transformar séries temporais em imagens; (C) Extração de *features*; (D) Classificação.

A. Processamento dos dados

Os dados utilizados para os experimentos do trabalho foram obtidos por meio da observação do tráfego de uma rede de 1gb por 3 dias, na qual houveram cenários de ataques *DoS* para gerar instabilidades e possibilitar suas análises. Na Figura 1 é possível visualizar o recorte da série temporal em um período observado de temperaturas de uma sala.

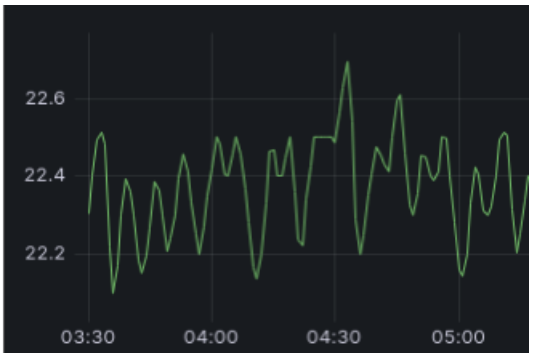

Fig. 1. Dados de séries temporais

Após a obtenção dos dados de série temporal, o período registrado foi divido em pedaços chamados *chunks*, no qual cada pedaço possui um tamanho de 100 observações. O número total de *chunks* criados é obtido pela Eq. (1). No qual a é a quantidade de dados observados, b o tamanho dos *chunks*.

$$\frac{a}{b} \tag{1}$$

Uma vez segmentado os dados, é possível realizar a transformação de séries temporais em imagens.

B. Transformação de séries temporais em imagens

Após a etapa de processamento dos dados, os *chunks* formados convertidos em imagens utilizando *Gramian Angular Summation Field* (*GASF*). Os dados de séries temporais são normalizados para que todos os valores estejam no intervalo [-1,1]. Em seguida, são calculadas as coordenadas polares e por fim são geradas as imagens [11].

Dado um vetor de série temporal $X = \{x_1, \ldots, x_n\}$ com n sendo o número de observações, para normalizar os dados de séries temporais no intervalo [-1,1] é aplicada a Eq. (2) .

$$\tilde{x}_i = \frac{(x_i - max(X)) + (x_i - min(X)}{max(X) - min(X)} \tag{2}$$

É possível representar os dados de séries temporais normalizados $\tilde{X}$ em um sistema de coordenadas polares ao codificar

TABLE I
TRABALHOS CORRELATOS

	(Xia et al, 2018	***(Mao et al, 2020***	***(Hussain et al, 2020***	***(Terzi, 2022***
Dados de invasões de redes/*softwares*	Sim	Não	Sim	Sim
Transforma séries temporais em imagens	Sim	Sim	Sim	Sim
Aplica *GAF* para a transformação	Não	Sim	Não	Sim
Utiliza *CNN* para extração de *features*	Não	Não	Sim	Sim
Utiliza *SVM* para classificar o modelo	Sim	Não	Não	Não

o valor como cosseno angular e a marcação temporal como raio, como mostrada na Eq. (3)

$$\begin{cases} \phi = \arccos(\tilde{x}_i), -1 \geq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{N}, t_i \in \mathbb{N} \end{cases} \tag{3}$$

Após a transformação dos dados normalizados no sistema de coordenadas polares, pode ser explorada a perspectiva angular ao considerar a soma trigonométrica entre cada ponto para identificar a correlação temporal em diferentes intervalos de tempo [9]. O *Gramian Angular Summation Field* pode ser definido na Eq. (4)

$$GASF = [\cos(\phi_i + \phi_j)] \tag{4}$$

Os dados de ataques *DoS* normalizados em coordenadas polares são expressados pela Matriz de Gram, G, expressada pela Eq. (5). A Matriz de Gram preserva os aspectos temporais dos dados desde que o *timestamp* aumente ao mover da posição (1,1) para (n,n) [9].

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix} \tag{5}$$

Um exemplo de imagem transformada pelo *Gramian Angular Summation Field* pode ser observada pela Fig. 2

C. Extração de features

Para a extração de *features* das imagens transformadas, foi utilizado o modelo de *CNN* chamado *VGG-16*.

A *VGG-16* é uma *deep CNN* desenvolvida pelo *Visual Geometry Group* (*VGG*) da Universidade de Oxford e é uma das redes mais populares na área de visão computacional. Os dados de entrada da *VGG-16* são imagens RGB com o tamanho fixo de 224 x 224, o único pré-processamento realizado é a subtração da média do valor RGB de cada pixel. A imagem passa por uma pilha de camadas convolucionais, nas quais utilizam filtros de tamanho 3x3, pois é o menor tamanho para capturar a noção de direita/esquerda/cima/baixo.

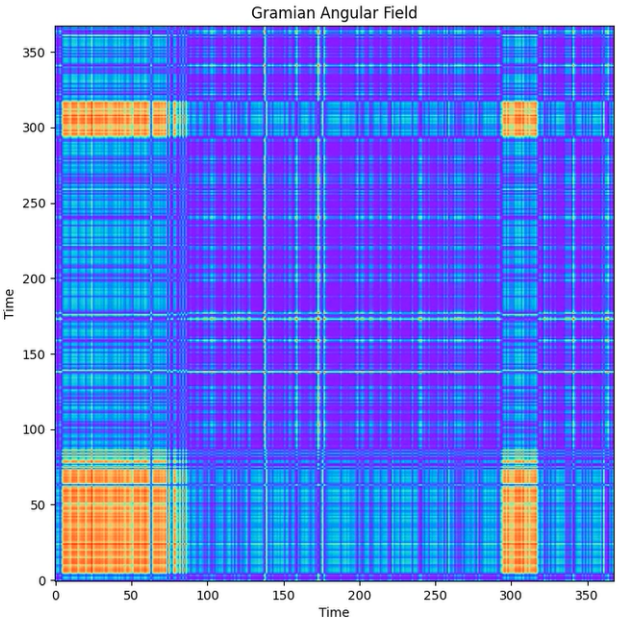

Fig. 2. Imagem transformada

O *pooling* espacial é realizado por cinco camadas de *max-pooling* que seguem algumas das camadas de convolução. O *max-pooling* é realizado em uma janela de 2x2 pixeis, com passo 2 [7].

Uma vez definida as técnicas utilizadas para a extração de *features*, é necessário um grande conjunto de imagens para gerar robustez para o modelo proposto e gerar métricas mais precisas. Para isto, foi aplicada uma geração de imagens que define várias transformações para aumentar os dados como rotações, translações, cisalhamento, zoom e inversão horizontal.

No processo de extração das *features* foram geradas 4 versões de cada imagem para realizar o aumento do número de dados, assim, possibilitou a aplicação dos modelos selecionados. As *features* extraídas pela *VGG-16* e adicionadas em um

vetor de características, além da criação de um vetor com um rótulo para imagens normais e imagens com ataques.

D. Classificação

O *Support Vector Machine* (*SVM*) é um algoritmo de aprendizado de máquina supervisionado para classificação e regressão. O *SVM* tem sido utilizado como uma ferramenta poderosa para resolver problemas práticos de classificação binária [2].

O modelo utilizado para classificação foi o *Support Vector Classifier* (*SVC*). Seu objetivo principal é encontrar um hiperplano em espaço de alta dimensão que separa pontos de diferentes classes com a maior margem possível. A margem é distância entre o hiperplano e os pontos de dados mais próximos de qualquer classe, chamados de vetores de suporte [8].

Na Fig. 3 é possível observar o fluxograma do método proposto, desde o processamento dos dados, a transformação em imagens, a extração de *features* e por fim a classificação.

IV. Resultados e discussões

Para demonstrar a viabilidade e eficiência do trabalho, um grande conjunto de dados é essencial para o treinamento do modelo. O *dataset* utilizado para o trabalho foi obtido por meio da observação do tráfego de uma rede com largura de 1GB com duração média de 3 dias, na qual houveram cenários de ataques *DoS* e, após a obtenção destes dados, foram transformados em imagens que, por sua vez, geraram mais imagens utilizando `ImageDataGenerator` da biblioteca *Tensorflow* a fim de aumentar o conjunto de dados. Neste contexto, foram utilizadas 2256 imagens para classificar o modelo e os dados foram divididos de maneira que 80% foram separados para treinamento e 20% foram separados para os testes. Na Tabela I é possível observar a divisão entre os dados utilizados para o modelo.

TABLE II
Divisão do dataset

Total	***Treinamento***	***Teste***
2256	1805	451

Na Figura 4 é possível observar dois exemplos de cada tipo de imagem, no qual as imagens (a) são geradas pelo tráfego normal de rede e as imagens (b) são as imagens geradas por ataques *DoS*.

A. Classification Report

Deste modo, foi realizado o experimento para classificação binária dos resultados por meio do *Classification Report* provido pelo classificador, as métricas utilizadas para verificar a confiabilidade do método proposto foram precisão, *Recall*, *F1-score* e acurácia. Na Tabela II é possível observar as métricas obtidas.

TABLE III
Resultados obtidos

Tipo de Imagem	**Métricas**		
	Precisão	***Recall***	***F1-score***
Normal	98%	94%	96%
Ataque	98%	99%	99%

B. Cross-Validation

Além do *Classification Report* provido pelo classificador, foi utilizado *Cross-Validation* com K_{folds} igual a 5, representados na Tabela III, com seus devidos *Scores* e sua média de *Score*.

TABLE IV
Resultados obtidos por Cross-Validation

Fold	1	2	3	4	5
Scores	0.988	0.976	0.982	0.980	0.8990
Mean Score	0.9836				

V. Conclusão

Durante o trabalho, foram exploradas abordagens para detecção de ataques de negativa de serviço (*DoS*) utilizando transformações de séries temporais em imagens via GAF e por conseguinte a extração de *features* por meio de uma CNN pré-treinada. A proposta se mostrou eficaz, alcançando acurácia de 98% na detecção de ataques *DoS*, demonstrando robustez na metodologia utilizada.

A transformação de series temporais em imagens permitiu a análise do tráfego de rede, facilitando a identificação de padrões anômalos, como os de ataques *DoS*. A utilização de uma CNN pré-treinada, especificamente o modelo VGG-16, provou ser uma escolha correta para a extração de *features*, capturando informações criticas para a detecção dos ataques.

Além disso, a utilização de técnicas para aumento de dados e a utilização do *SVM* em seu modelo classificador permitiu com que o modelo houvesse uma grande acurácia. Ademais, a utilização de *Cross-Validation* confirmou a consistência dos resultados demonstrado na Tabela II.

Portanto, a abordagem mostra-se promissora para a detecção de ataques *DoS*, e também abre caminho a novas pesquisas relacionadas ao tema, como: Detecção de *DDoS*, estudo de modelos alternativos como *XGBoost* e outros modelos de ataques mais avançados.

References

[1] Bindu Bala and Sunny Behal. Ai techniques for iot-based ddos attack detection: Taxonomies, comprehensive review and research challenges. *Computer science review*, 52:100631, 2024.

[2] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[3] Fortinet. Fortinet relata que a américa latina foi alvo de mais de 360 bilhões de tentativas de ataques cibernéticos em 2022.

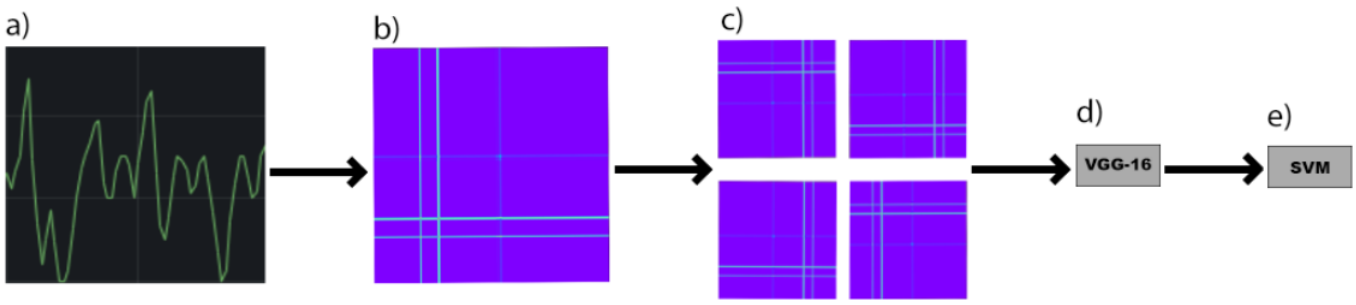

Fig. 3. Dados de séries temporais (a), Imagens transformadas (b), Imagens aleatórias geradas (c), *CNN* (d) e classificador (e)

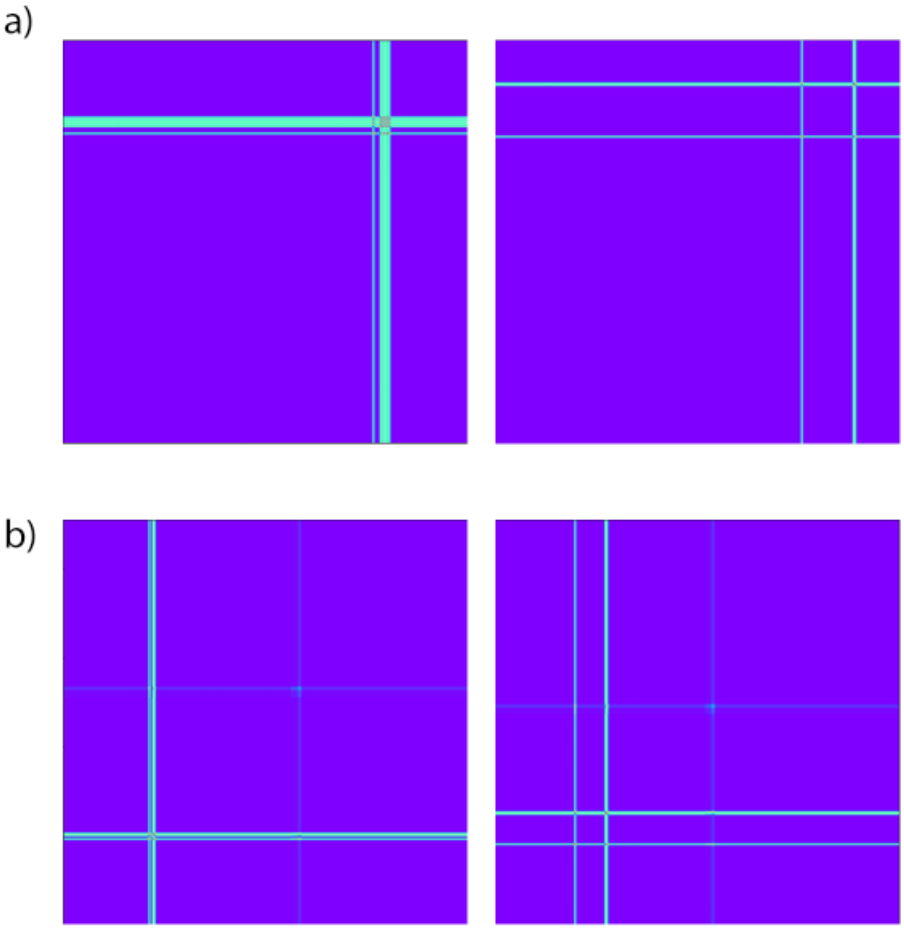

Fig. 4. Imagens *GAF* para tráfego normal (a), Ataques *DoS* (b)

[4] Faisal Hussain, Syed Ghazanfar Abbas, Muhammad Husnain, Ubaid U Fayyaz, Farrukh Shahzad, and Ghalib A Shah. Iot dos and ddos attack detection using resnet. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pages 1–6. IEEE, 2020.

[5] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

[6] Jianxiao Mao, Hao Wang, and Billie F Spencer Jr. Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders. *Structural Health Monitoring*, 20(4):1609–1626, 2021.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] Madan Somvanshi, Pranjali Chavan, Shital Tambade, and S. V. Shinde. A review of machine learning techniques using decision tree and support vector machine. In *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, pages 1–7, 2016.

[9] Duygu Sinanc Terzi. Gramian angular field transformation-based intrusion detection. *Computer Science*, 23, 2022.

[10] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[11] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. *arXiv preprint arXiv:1506.00327*, 2015.

[12] Shiming Xia, Zhisong Pan, Zhe Chen, Wei Bai, and Haimin Yang. Malware classification with markov transition field encoded images. In *2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pages 1–5. IEEE, 2018.