

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE SÃO JOÃO DA BOA VISTA

AMANDA CRISTINA NOVAES

**Estudo sobre o desempenho da aplicação de máquinas de vetores de suporte em sistemas ópticos
coerentes digitais.**

São João da Boa Vista
2024

Amanda Cristina Novaes

Estudo sobre o desempenho da aplicação de máquinas de vetores de suporte em sistemas ópticos coerentes digitais.

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Eletrônica e de Telecomunicações do Campus de São João da Boa Vista, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Eletrônica e de Telecomunicações .

Orientador: Profº Dr. Ivan Aritz Aldaya Garde

São João da Boa Vista

2024

N935e

Novaes, Amanda Cristina

Estudo sobre o desempenho da aplicação de máquinas de vetores de suporte em sistemas ópticos coerentes digitais. / Amanda Cristina

Novaes. -- São João da Boa Vista, 2024

45 p.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Telecomunicações) - Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, São João da Boa Vista

Orientador: Ivan Aritz Aldaya Garde

1. Inteligência artificial. 2. Comunicações ópticas. 3. Fibras ópticas.
4. Telecomunicação. I. Título.

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA - CÂMPUS DE SÃO JOÃO DA BOA VISTA
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES**

TRABALHO DE CONCLUSÃO DE CURSO

**ESTUDO SOBRE O DESEMPENHO DA APLICAÇÃO DE MAQUINAS DE
VETORES DE SUPORTE EM SISTEMAS ÓPTICOS COERENTES DIGITAIS**

Aluno: Amanda Cristina Novaes
Orientador: Prof. Dr. Ivan Aritz Aldaya Garde

Banca Examinadora:

- Ivan Aritz Aldaya Garde (Orientador)
- Cintya Wink de Oliveira Benedito (Examinadora)
- Marlon Rodrigues Garcia (Examinador)

Os formulários de avaliação e a ata da defesa, na qual consta a aprovação do trabalho, devidamente assinados pela banca encontram-se no prontuário eletrônico do aluno.

São João da Boa Vista, 04 de dezembro de 2024

RESUMO

O avanço nos sistemas de comunicações ópticas tem permitido aumentos significativos nas taxas de transmissão de dados, especialmente com o desenvolvimento dos sistemas ópticos coerentes digitais. A capacidade de modulação do sinal em amplitude, fase e polarização viabiliza transmissões na ordem de centenas de gigabits por segundo de maneira eficiente. No entanto, essas altas taxas de transmissão tornam imprescindível a mitigação de efeitos lineares e não lineares. Este estudo utilizou um método baseado em algoritmos de inteligência artificial para compensar as não linearidades em uma rede óptica passiva coerente digital de longo alcance.

A transmissão de um sinal óptico com dupla polarização e modulação de amplitude em quadratura de 16 níveis (16-QAM - *Quadrature Amplitude Modulation*) foi simulada computacionalmente utilizando o software *VPI Transmission Maker* em um enlace de 140 km, considerando os impactos de efeitos lineares e não lineares. Fenômenos como ruído e atenuação foram identificados como os principais fatores limitantes no alcance dos sistemas ópticos. Embora o aumento da potência de transmissão seja, à primeira vista, uma estratégia atrativa para melhorar a relação sinal-ruído, esse aumento pode intensificar as não linearidades devido à interação do sinal com o meio de propagação. Para mitigar os efeitos não lineares, foi empregado o método de regressão por vetores de suporte. A implementação foi realizada utilizando a linguagem de programação *Python* no ambiente *Jupyter Notebook*.

Os resultados obtidos demonstraram que algoritmos baseados em máquinas de vetores de suporte são eficazes na compensação das não linearidades, apresentando uma melhora na taxa de erro de bit em diferentes taxas de transmissão. Ao comparar o algoritmo de classificação por vetores de suporte com o algoritmo de regressão por vetores de suporte, observou-se que o algoritmo de classificação apresentou um desempenho ligeiramente melhor, especialmente para potências de transmissão superiores a 9 dBm.

PALAVRAS-CHAVE: Inteligência artificial. Comunicações ópticas. Máquinas de vetores de suporte. Regressão de vetores de suporte.

ABSTRACT

The advancements in optical communication systems have enabled significant increases in data transmission rates, particularly with the development of digital coherent optical systems. The ability to modulate signals in amplitude, phase, and polarization facilitates transmissions at rates in the order of hundreds of gigabits per second efficiently. However, these high transmission rates make it essential to mitigate linear and nonlinear effects. This study employed a method based on artificial intelligence algorithms to compensate for nonlinearities in a long-haul digital coherent passive optical network. The transmission of an optical signal with dual polarization and 16-level quadrature amplitude modulation (16-QAM) was computationally simulated using the *VPI Transmission Maker* software over a 140 km link, considering the impacts of linear and nonlinear effects. Phenomena such as noise and attenuation were identified as the main limiting factors in the reach of optical systems. Although increasing transmission power might initially appear to be an attractive strategy to improve the signal-to-noise ratio, this increase can intensify nonlinearities due to the interaction of the signal with the propagation medium. To mitigate nonlinear effects, a support vector regression method was employed. The implementation was carried out using the Python programming language in the *Jupyter Notebook* environment.

The results obtained demonstrated that algorithms based on support vector machines are effective in compensating for nonlinearities, showing an improvement in the bit error rate at different transmission rates. When comparing the support vector classification algorithm with the support vector regression algorithm, it was observed that the classification algorithm performed slightly better, especially for transmission powers above 9 dBm.

KEYWORDS: Artificial intelligence. Optical communications. Support vector machines. Support vector regression.

LISTA DE ILUSTRAÇÕES

Figura 1	Demonstração de um caso de classificação com hiperplano separador ótimo . .	18
Figura 2	Demonstração de um caso de SVR	19
Figura 3	Diagrama de blocos da configuração da simulação. DAC: Conversor digital-analógico (<i>Digital-to-Analog Converter</i>), DA: Amplificador de driver (<i>Driver Amplifier</i>), LD: Laser de diodo (<i>Diode Laser</i>), PBS: Divisor de feixe de polarização (<i>Polarization Beam Splitter</i>), DP-MZM: Modulador Mach–Zehnder duplo-paralelo (<i>Dual-Parallel Mach–Zehnder Modulator</i>), PBC: Combinador de feixe de polarização (<i>Polarization Beam Combiner</i>), EDFA: Amplificador de fibra dopada com érbio (<i>Erbium-Doped Fiber Amplifier</i>), SMF: Fibra monomodo (<i>Single-Mode Fiber</i>), PD: Fotodetector (<i>photodetector</i>) e ADC: Conversor analógico-digital (<i>Analog-to-Digital Converter</i>).	22
Figura 4	Diagrama de constelação para potências de transmissão de 1 a 12dBm	23
Figura 5	Diagrama de Constelação para SP e potência de transmissão 12 dBm	24
Figura 6	Diagrama de Constelação para DP e potência de transmissão 12 dBm	25
Figura 7	BER em função da potência óptica de transmissão considerando processamento DP e SP.	26
Figura 8	BER em função da potência óptica considerando SP e diferentes valores do parâmetro C.	27
Figura 9	BER em função da potência óptica considerando DP e diferentes valores do parâmetro C.	27
Figura 10	BER em função da potência óptica considerando SP e diferentes valores de γ . .	28
Figura 11	BER em função da potência óptica considerando DP e diferentes valores de γ . .	29
Figura 12	Comparação SVR, SVC e ML	30

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog-to-Digital Converter</i> (Conversor analógico-digital)
AI	<i>Artificial Intelligence</i> (Inteligência artificial)
ASK	<i>Amplitude Shift Keying</i> (Modulação por chaveamento de amplitude)
BER	<i>Bit Error Rate</i> (Taxa de erro de bits)
DAC	<i>Digital-to-Analog Converter</i> (Conversor digital-analógico)
DA	<i>Driver Amplifier</i> (Amplificador de driver)
DBSCAN	<i>Density-Based Spatial Clustering of Applications with Noise</i> (Clusterização baseada em densidade com ruído)
DP	<i>Dual Polarization</i> (Polarização dupla)
DP-MZM	<i>Dual-Parallel Mach–Zehnder Modulator</i> (Modulador Mach–Zehnder duplo-paralelo)
DSP	<i>Digital Signal Processor</i> (Processador de sinais digitais)
DWDM	<i>Dense Wavelength Division Multiplexing</i> (Multiplexação por divisão densa de comprimento de onda)
EDFA	<i>Erbium-Doped Fiber Amplifier</i> (Amplificador de fibra dopada com érbio)
FSK	<i>Frequency Shift Keying</i> (Modulação por chaveamento de frequência)
FWM	<i>Four Wave Mixing</i> (Mistura de quatro ondas)
LD	<i>Diode Laser</i> (Laser de diodo)
LED	<i>Light-Emitting Diode</i> (Diodo emissor de luz)
ML	<i>Machine Learning</i> (Aprendizado de máquina)
ML	<i>Maximum likelihood</i> (Máxima verossimilhança)
PBC	<i>Polarization Beam Combiner</i> (Combinador de feixe de polarização)
PBS	<i>Polarization Beam Splitter</i> (Divisor de feixe de polarização)
PCA	<i>Principal Component Analysis</i> (Análise de componentes principais)
PD	<i>Photodetector</i> (Fotodetector)
PRBS	<i>Pseudo Random Binary Sequence</i> (Sequência binária pseudoaleatória)

PSK	<i>Phase Shift Keying</i> (Modulação por chaveamento de fase)
QAM	<i>Quadrature Amplitude Modulation</i> (Modulação em amplitude de quadratura)
RBF	<i>Radial Basis Function</i> (Função de base radial)
SARSA	<i>State-Action-Reward-State-Action</i> (Estado-ação-recompensa-estado-ação)
SBS	<i>Stimulated Brillouin Scattering</i> (Espalhamento estimulado de Brillouin)
SMF	<i>Single Mode Fiber</i> (Fibra monomodo)
SP	<i>Single Polarization</i> (Polarização única)
SPM	<i>Self Phase Modulation</i> (Automodulação de fase)
SRS	<i>Stimulated Raman Scattering</i> (Espalhamento estimulado de Raman)
SVC	<i>Support Vector Classifier</i> (Classificação por vetores de suporte)
SVR	<i>Support Vector Regression</i> (Regressão por vetores de suporte)
SVM	<i>Support Vector Machine</i> (Máquinas de vetores de suporte)
WDM	<i>Wavelength Division Multiplexing</i> (Multiplexação por divisão de comprimento de onda)
XPM	<i>Cross Phase Modulation</i> (Modulação de fase cruzada)

LISTA DE SÍMBOLOS

C	Coeficiente de regularização do modelo SVR
ϵ	Parâmetro de insensibilidade ao erro no modelo SVR
$f(x)$	Função preditiva no modelo SVR
γ	Parâmetro de influência dos vetores de suporte no modelo SVR
s	Segundos
Mb	Megabit
Gb	Gigabit
nm	Nanômetro
mW	Miliwatt
μm	Micrômetro
km	Quilômetro
GSa	Gigasample
dB	Decibel

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Contextualização histórica	10
1.2	Sistemas ópticos coerentes digitais	12
1.2.1	Efeitos não lineares na transmissão por fibra óptica	13
1.3	Descrição do problema e contribuição do trabalho	14
1.4	Organização do documento	14
2	CLASSIFICAÇÃO E REGRESSÃO COM MÁQUINAS DE VETORES DE SUPORTE	16
2.1	Aprendizado de máquina supervisionado e não supervisionado	16
2.1.1	Aprendizado supervisionado	16
2.1.2	Aprendizado não supervisionado	17
2.1.3	Aprendizado por reforço	17
2.2	Máquinas de Vetores de Suporte	18
2.3	Regressão mediante máquinas de vetores de suporte	19
3	CONFIGURAÇÕES DA SIMULAÇÃO E DO SVR	21
3.1	Configurações da simulação	21
3.2	Configuração do SVR	22
4	RESULTADOS	23
4.1	Análise de desempenho	23
4.2	Comparação com um sistema SVC	29
5	CONCLUSÃO	31
	REFERÊNCIAS	32
	APÊNDICE A – CÓDIGO EM PYTHON	34
A.1	Compensação de efeitos não lineares utilizando SVR	34

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO HISTÓRICA

Um sistema de comunicação óptica é aquele que usa ondas eletromagnéticas no espectro óptico como portadora dos sinais, abrangendo frequências de cerca de 0,05 até 100 μm . Estes sistemas começaram a ser estudados na década de 60 com a invenção do Laser, e foram essas pesquisas iniciais que levaram ao formato moderno de comunicações ópticas que conhecemos hoje (NUNES, 2001).

Nos últimos anos, o crescimento exponencial do número de usuários de internet e da quantidade de dispositivos conectados por indivíduo tem impulsionado significativamente a demanda por serviços de telecomunicações, exigindo a expansão e o aumento da capacidade das redes. Nesse contexto, os sistemas de comunicações ópticas desempenham um papel estratégico, consolidando-se cada vez mais como a base das principais operadoras de telecomunicações em escala global (PINTO, 1999).

Desde o início dos estudos sobre comunicações ópticas na década de 1970, surgiram cinco gerações distintas desses sistemas, cada uma proporcionando avanços significativos tanto na velocidade de transmissão quanto na distância alcançada (AGRAWAL, 2021).

- **Primeira geração:** A primeira geração de sistemas de comunicações ópticas foi desenvolvida no início dos anos 70 e operava a uma taxa de 45 Mb/s. Naquela época, mesmo essa taxa relativamente baixa, representava uma grande inovação em comparação com as redes de fios de cobre existentes (AGRAWAL, 2021).

Os diodos emissores de luz (LED - *Light-Emitting Diode*) utilizados para a transmissão de sinais nessa geração emitiam cerca de 1 mW de radiação eletromagnética com uma largura espectral na faixa de 800 nm. A intensidade da luz era ajustada por meio da modulação da corrente de injeção do LED. A transmissão do sinal óptico do emissor até o receptor, composto por um fotodíodo de silício, era realizada através de fibra multimodo. No entanto, devido às perdas relativamente altas na fibra óptica, cerca de 2,5 dB/km, a distância máxima de transmissão entre repetidores era de 10km (KEISER, 1991).

- **Segunda Geração:** A segunda geração de sistemas de comunicações ópticas começou no final dos anos 70 e trouxe melhorias tanto na taxa de transmissão de bits quanto no espaçamento entre repetidores. Com a introdução de lasers semicondutores e fotodetectores na faixa de 1300 nm, onde as perdas da fibra eram em torno de 1 dB/km, a taxa de transmissão dos sistemas chegou a 100 Mb/s, mas estava limitada devido à dispersão intermodal. Esse problema foi solucionado no início dos anos 80 com o desenvolvimento das fibras monomodo (AGRAWAL, 2021).

Com as fibras monomodo, os efeitos de dispersão foram reduzidos, pois esse tipo de fibra permite a propagação em um único modo, eliminando a dispersão intermodal. Como resultado, os sistemas passaram a operar a uma taxa de 1,7 Gb/s com um espaçamento entre repetidores de 50 km e perdas da ordem de 0,6 dB/km operando na faixa de 1300 nm. Porém o espaçamento entre repetidores ainda era limitado pela atenuação da fibra (PINTO, 1999).

- **Terceira Geração:** Na terceira geração de sistemas de comunicações ópticas, passaram a ser utilizados fontes e detectores operando a 1550 nm. A atenuação da fibra de sílica fundida é extremamente baixa nesse comprimento de onda, em torno de 0,2 dB/km. No entanto, a dispersão cromática relativamente alta nesse comprimento de onda resultou em atrasos no desenvolvimento dos sistemas dessa geração (PINTO, 1999).

Para enfrentar o desafio da dispersão, foram adotadas duas estratégias principais: a primeira envolveu o desenvolvimento de lasers monomodo, ou seja, lasers que transmitem um único comprimento de onda, e a segunda, o aperfeiçoamento de fibras com deslocamento de dispersão. A combinação dessas duas soluções proporcionou um desempenho significativamente melhor e esses aprimoramentos permitiram taxas de transmissão de até 10 Gb/s, mais de 200 vezes superiores às dos sistemas iniciais, e expandiram a distância de alcance para 100 km sem a necessidade de regeneração de sinal (AGRAWAL, 2021).

Apesar desses avanços, uma limitação dos sistemas da terceira geração era a necessidade de regeneração eletrônica. Embora métodos de detecção coerente tenham sido explorados para melhorar a sensibilidade do receptor, essa abordagem foi eventualmente substituída pelo desenvolvimento de amplificadores ópticos, que se tornou um elemento chave na próxima geração de sistemas ópticos (AGRAWAL, 2021).

- **Quarta Geração:** Os sistemas de quarta geração se basearam no uso de amplificadores ópticos para aumentar o espaçamento entre repetidores, e na multiplexação por divisão de comprimento de onda (WDM - *Wavelength Division Multiplexing*) para ampliar a taxa de bits agregada (AGRAWAL, 2021).

Durante os anos 80, foram desenvolvidos amplificadores de fibra dopada com érbio para amplificar sinais sem necessidade de regeneração eletrônica. Esses amplificadores utilizam fibra dopada com íons de érbio, que são excitados com lasers de bombeio na faixa de 980 nm ou 1480 nm. Quando os íons excitados retornam ao seu estado original, eles emitem luz na faixa de 1500 nm, amplificando assim os sinais através da emissão estimulada (BENINCA, 2012). Os amplificadores ópticos atuam diretamente na amplificação dos sinais de luz, superando a perda sem a necessidade de conversão óptica-eletrônica-óptica. Com isso, as conexões de fibra puderam ser estendidas por mais de 10.000 km sem a necessidade de regeneração eletrônica do sinal (PINTO, 1999).

Ao combinar esses amplificadores com a tecnologia WDM, que permite a transmissão de sinais em diferentes comprimentos de onda em uma única fibra monomodo, foi possível amplificar múltiplos comprimentos de onda usando o mesmo amplificador óptico (AGRAWAL, 2021).

- **Quinta geração:** À medida que avançamos para os anos 2000, a capacidade das redes de fibra óptica experimentou uma expansão significativa (AGRAWAL, 2021). A quinta geração levou essa capacidade a novos patamares com a introdução da multiplexação por divisão de comprimento de onda densa (DWDM - *Dense Wavelength Division Multiplexing*). Enquanto os primeiros sistemas WDM ofereciam apenas de 4 a 8 canais, os sistemas DWDM de última

geração foram capazes de utilizar até 160 canais, cada um transportando sinais com taxas de 40 a 100 Gb/s (PINTO, 1999).

Além dessa ampliação na capacidade de multiplexação, também surgiram os sistemas ópticos de detecção coerente nesta geração. Embora a detecção coerente já fosse utilizada em comunicações de rádio desde a década de 1930, a introdução de lasers monomodo com larguras espectrais muito estreitas e baixo ruído de fase possibilitou a aplicação dessas técnicas também em sistemas ópticos (PINTO, 1999).

Com a aplicação dos sistemas ópticos coerentes digitais, tornou-se possível a extração de mais informações do sinal, aumentando a tolerância a problemas ópticos de comunicação e, consequentemente, melhorando a performance do sistema (BARBOSA, 2021).

1.2 SISTEMAS ÓPTICOS COERENTES DIGITAIS

Em paralelo ao avanço dos sistemas de comunicações ópticas, a eletrônica também evoluiu significativamente, especialmente no campo da eletrônica de altas frequências. Isso possibilitou o processamento digital de sinais (DSP - *Digital Signal Processing*) em tempo real para sinais com taxas da ordem de Gigaamostras por segundo (GSa/s) (BORGES, 2021).

A partir desse ponto, vários grupos de pesquisa começaram a explorar a utilização desses processadores digitais de alta velocidade para compensar efeitos que dificultavam a detecção coerente, como o ruído de fase e a flutuação no estado de polarização. Com esses avanços, tornou-se viável a implementação de sistemas ópticos com recepção coerente digital (BORGES, 2021).

Uma das principais vantagens da recepção coerente digital é a capacidade de converter todas as características do sinal óptico para o domínio elétrico, permitindo seu processamento no domínio digital. Diferentemente das tecnologias tradicionais, a detecção coerente não se restringe à medição da amplitude do sinal de luz, mas também inclui informações sobre a fase e a polarização. Esse método proporciona melhorias significativas na sensibilidade e na eficiência espectral do receptor óptico, viabilizando a transmissão de maiores volumes de dados por distâncias mais longas (ROUDAS, 2011).

Por meio do processamento digital de sinais, aliado à habilidade de detectar e processar a fase e a polarização do sinal de luz, esse sistema consegue corrigir problemas ópticos de transmissão, como a dispersão cromática e a dispersão do modo de polarização (ROUDAS, 2011). Ademais, o uso de algoritmos de processamento digital possibilita a compensação da diferença de frequência entre o oscilador local e o sinal, de maneira totalmente digital (DINIZ, 2013).

Além disso, a aplicação de detecção coerente em sistemas ópticos também possibilitou a utilização de diversas técnicas de modulação na transmissão dos sinais, tais como modulação digital de fase (PSK - *Phase Shift Keying*), modulação digital de frequência (FSK - *Frequency Shift Keying*) e modulação digital de amplitude (ASK - *Amplitude Shift Keying*). Essas técnicas permitem a exploração da diversidade de fase e polarização dos sinais (PINTO, 1999).

Porém, apesar da aplicação da recepção coerente ter permitido a correção de diversos problemas presentes nos sistemas de comunicações ópticas, a presença de efeitos não lineares nestes sistemas

podem levar a erros na interpretação dos dados (AGRAWAL, 2021).

1.2.1 Efeitos não lineares na transmissão por fibra óptica

Em sistemas de comunicações ópticas, a maior parte dos efeitos não lineares são provenientes das fibras ópticas. O uso deste meio de transmissão em sistemas de longa distância exige potências de transmissão mais elevadas e modulações mais complexas do que em sistemas de curta distância. Essa necessidade acarreta efeitos não lineares devido à interação entre os campos elétricos intensos e o dielétrico da fibra, geralmente de sílica (AGRAWAL, 2007).

Esses efeitos não lineares são considerados a maior barreira em sistemas de longa distância, pois podem alterar significativamente os pulsos transmitidos, tornando impossível recuperar a informação enviada. Os principais impedimentos não lineares incluem o espalhamento estimulado de luz e o efeito Kerr (MIRANDA, 2011).

- **Espalhamento estimulado de luz:** O espalhamento estimulado de luz é um fenômeno que ocorre quando a luz é modulada por vibrações, resultando em um desvio de frequência. Este efeito pode ser subdividido em espalhamento estimulado de Brillouin (SBS - *Stimulated Brillouin Scattering*) e espalhamento estimulado de Raman (SRS - *Stimulated Raman Scattering*).

O SBS ocorre quando a potência óptica é suficientemente alta para que a interação entre a luz e as moléculas da fibra gere vibrações acústicas. Essas vibrações dão origem a uma onda que se propaga na direção oposta à transmissão, atuando como uma forma de atenuação não linear. Caso a potência ultrapasse um determinado limiar, o processo se torna estimulado, fazendo com que uma parte significativa do sinal seja refletida, criando uma malha de realimentação positiva que amplifica a onda espalhada (AGRAWAL, 2007).

De forma semelhante ao SBS, o SRS também resulta da interação entre a luz e os modos vibracionais das moléculas de sílica. No entanto, nesse caso, são gerados fótons que causam o espalhamento da luz, tanto nas direções propagante quanto co-propagante. Durante este processo, a energia luminosa é absorvida pelas moléculas, excitando seus elétrons, que, ao retornarem a níveis de energia estáveis, liberam fótons. Esse processo faz com que a luz seja espalhada em frequências diferentes das originais (MIRANDA, 2011).

- **Efeito Kerr:** O efeito Kerr ocorre devido à interação entre a luz e o material da fibra óptica, provocando alterações não lineares no índice de refração. Essas alterações dependem da potência do feixe e da área efetiva da fibra. Como resultado, há um deslocamento de fase não linear que pode se manifestar como automodulação de fase (SPM - *Self Phase Modulation*), onde a fase do sinal varia devido às mudanças temporais na potência do pulso, ou como modulação de fase cruzada (XPM - *Cross Phase Modulation*), onde a fase é influenciada pela intensidade de sinais copropagantes em sistemas WDM. Além disso, o fenômeno de mistura de quatro ondas (FWM - *Four Wave Mixing*) ocorre quando duas ou três ondas transferem energia entre si, criando novas frequências com base nos princípios de conservação de energia e momento (AGRAWAL, 2007).

De modo geral, modulação de fase cruzada e mistura de quatro ondas são efeitos que ocorrem exclusivamente quando se faz uso de múltiplos canais, como é o caso de sistemas WDM,

enquanto que a automodulação de fase ocorre em qualquer tipo de sistema de comunicação óptica (AGRAWAL, 2007).

1.3 DESCRIÇÃO DO PROBLEMA E CONTRIBUIÇÃO DO TRABALHO

Os sistemas de comunicações ópticas coerentes digitais oferecem diversas vantagens, como alta velocidade, ampla largura de banda, baixa atenuação e imunidade a interferências eletromagnéticas. Contudo, esses sistemas também apresentam limitações significativas. Em particular, os efeitos não lineares nas fibras ópticas causam distorções no sinal, restringindo o desempenho. Apesar da existência de várias técnicas para mitigar esses efeitos, a alta complexidade e os desafios de implementação prática têm dificultado sua aplicação.

Este estudo busca explorar os benefícios da aplicação de técnicas de Inteligência Artificial (AI - *Artificial Intelligence*) para reduzir os impactos dos efeitos não lineares em sistemas de comunicações ópticas coerentes digitais. Desta forma, o objetivo principal é utilizar regressão por vetores de suporte (SVR - *Support Vector Regression*), que é uma adaptação para regressão do método de máquinas de vetores de suporte (SVM - *Support Vector Machine*), em um sistema óptico coerente digital, buscando avaliar seu impacto na mitigação de efeitos lineares e não lineares por meio da taxa de erro de bit (BER - *Bit Error Rate*), e comparar seus resultados com os obtidos a partir da aplicação de máquinas de vetores de suporte para classificação (SCV - *Support Vector Classifier*) disponíveis em (ALDAYA et al., 2023).

A implementação do SVR foi realizada no ambiente *Jupyter Notebook*, utilizando a linguagem *Python*, enquanto a simulação do sistema óptico foi conduzida no software *VPI Transmission Maker*. O estudo analisa um sistema de comunicação óptica coerente de longo alcance com dupla polarização e modulação de amplitude em quadratura de 16 níveis (16-QAM - *Quadrature Amplitude Modulation*), considerando um enlace de 140 km. A mitigação das não linearidades foi avaliada processando cada polarização de forma independente e também considerando o processamento simultâneo das duas polarizações. A potência inserida na fibra foi analisada em um intervalo de 1 a 12 dBm, abrangendo diferentes cenários.

1.4 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho está organizado da seguinte forma:

Capítulo 1: Apresenta a evolução dos sistemas de comunicações ópticas até o cenário atual, com ênfase nos sistemas ópticos coerentes digitais, e discute os principais efeitos não lineares que impactam esses sistemas. Após essa contextualização, são apresentadas a justificativa do estudo e a contribuição do trabalho.

Capítulo 2: Apresenta o conceito de aprendizado de máquina supervisionado e não supervisionado, explorando suas diferenças e diversas aplicações práticas. O capítulo encerra com a introdução às técnicas de SVM.

Capítulo 3: Descreve a metodologia utilizada neste trabalho, incluindo o detalhamento da configuração de simulação e os ajustes dos parâmetros utilizados no modelo de SVR.

Capítulo 4: Apresenta os resultados obtidos nas simulações realizadas utilizando SVR, além de uma análise comparativa entre as metodologias de Classificação e Regressão em SVM.

Capítulo 5: Conclui o trabalho com um resumo das principais descobertas e conclusões derivadas dos resultados apresentados.

2 CLASSIFICAÇÃO E REGRESSÃO COM MÁQUINAS DE VETORES DE SUPORTE

2.1 APRENDIZADO DE MÁQUINA SUPERVISIONADO E NÃO SUPERVISIONADO

O aprendizado de máquina (ML - *Machine Learning*) é um subcampo da inteligência artificial que se concentra no desenvolvimento de algoritmos capazes de identificar padrões e extrair informações úteis a partir de grandes volumes de dados, sem a necessidade de programação explícita para cada tarefa. Esses algoritmos são projetados para aprender de maneira autônoma, aprimorando seu desempenho à medida que expostos a novos dados. Atualmente, o ML desempenha um papel importante em diversas aplicações, como reconhecimento de fala, diagnóstico médico, previsão de comportamento do usuário, além de seu uso em redes de comunicação óptica para mitigar efeitos não lineares e otimizar a transmissão de dados (AWAD; KHANNA, 2015).

As técnicas de ML permitem a automação de tarefas complexas, além de melhorar previsões e decisões com base em dados dinâmicos. Essas características fazem com que o aprendizado de máquina seja amplamente utilizado em uma grande variedade de problemas. O aumento significativo da capacidade de processamento de dados, aliado à diminuição dos custos computacionais, tem acelerado a adoção dessas técnicas (ALPAYDIN, 2014).

O aprendizado de máquina está dividido em três principais abordagens: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço, cada uma apropriada para diferentes tipos de problemas e cenários (YE, 2023).

2.1.1 Aprendizado supervisionado

No aprendizado supervisionado, os algoritmos são treinados com dados previamente rotulados, ou seja, são dados de entradas cujas saídas corretas já são conhecidas. Dessa forma, o algoritmo pode mapear a correlação existente entre as entradas e as saídas conhecidas, e assim, generalizando a partir dos dados de treinamento, o algoritmo é capaz de prever os rótulos para novas entradas não rotuladas. Esse método pode ser aplicado a tarefas de classificação e regressão (AWAD; KHANNA, 2015).

- **Classificação:** O objetivo da classificação é categorizar os dados de entrada em classes discretas predefinidas. Por exemplo, em um sistema de detecção de fraudes, o algoritmo precisa classificar transações como fraudulentas ou não. Alguns dos algoritmos mais usados incluem: classificação por vetores de suporte, redes neurais artificiais e árvores de decisão (ALPAYDIN, 2014).
- **Regressão:** A regressão é usada quando o objetivo é prever um valor contínuo. Um exemplo clássico seria prever o valor de uma ação na bolsa de valores com base em dados históricos. Os modelos mais comuns de regressão incluem a regressão linear e a regressão por vetores de suporte (ALPAYDIN, 2014).

Um ponto de atenção no aprendizado supervisionado é o sobreajuste, um problema que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento, capturando ruídos ou peculiaridades específicas, mas falhando em generalizar para novos dados. Para evitar esse problema, técnicas

como regularização e validação cruzada são empregadas, garantindo que o modelo mantenha sua capacidade preditiva ao lidar com dados não vistos (SHALEV-SHWARTZ; BEN-DAVID, 2014).

2.1.2 Aprendizado não supervisionado

Ao contrário do aprendizado supervisionado, o aprendizado não supervisionado lida com dados não rotulados. Em vez de utilizar rótulos, ele identifica as semelhanças entre os diferentes dados de entrada para categorizá-los ou determinar uma melhor representação deles. O objetivo é descobrir padrões ou estruturas intrínsecas nos dados explorados, sem qualquer informação prévia sobre os rótulos dos mesmos. Esse método é especialmente útil em cenários com grandes volumes de dados, uma vez que elimina a necessidade de etapas de rotulação, que podem gerar custos elevados em tempo de trabalho humano, e também dispensa as fases de treinamento, que apresentam alta demanda computacional (AWAD; KHANNA, 2015).

As duas principais técnicas no aprendizado não supervisionado são *clusterização* e redução de dimensionalidade:

- **Clusterização:** Algoritmos de *clusterização* agrupam dados semelhantes em subconjuntos chamados *clusters*. Esta técnica é particularmente útil em áreas como marketing, onde é usada para segmentar clientes com base em seus comportamentos, de modo que elementos dentro de um cluster sejam mais semelhantes entre si do que com elementos de outros *clusters*. Alguns exemplos de *clusterização* são: K-Means, DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) e maximização da expectativa (ALPAYDIN, 2014).
- **Redução de dimensionalidade:** Em muitos conjuntos de dados, existem variáveis redundantes ou dados irrelevantes que podem ser removidas sem perda significativa de informação. Técnicas como a análise de componentes principais são usadas para projetar os dados em um espaço de menor dimensão, preservando as características mais relevantes, o que é particularmente útil para visualização de dados ou para acelerar algoritmos (MAATEN; POSTMA; HERIK, 2009).

O aprendizado não supervisionado é uma área em crescimento, especialmente em contextos onde a rotulagem manual de dados é inviável ou custosa, como em grandes bases de dados de redes sociais ou dados genômicos (ALPAYDIN, 2014).

2.1.3 Aprendizado por reforço

A aprendizagem por reforço situa-se entre a aprendizagem supervisionada e a não supervisionada. Essa é uma abordagem na qual um agente aprende a tomar decisões interagindo com um ambiente. Ao contrário do aprendizado supervisionado, o agente não recebe instruções explícitas sobre a ação correta a tomar, mas aprende por meio de respostas, positivas ou negativas, a cada ação tomada. O objetivo do agente é maximizar as respostas positivas ao longo do tempo. Este paradigma é amplamente utilizado em robótica, sistemas de controle, e até em jogos (AWAD; KHANNA, 2015).

Alguns algoritmos comuns de aprendizado por reforço são o Q-Learning e o SARSA (*State-Action-Reward-State-Action*). O aprendizado por reforço é crucial em cenários onde o aprendizado

não ocorre a partir de um conjunto fixo de dados, mas de experiências e interação contínua com o ambiente (ALPAYDIN, 2014).

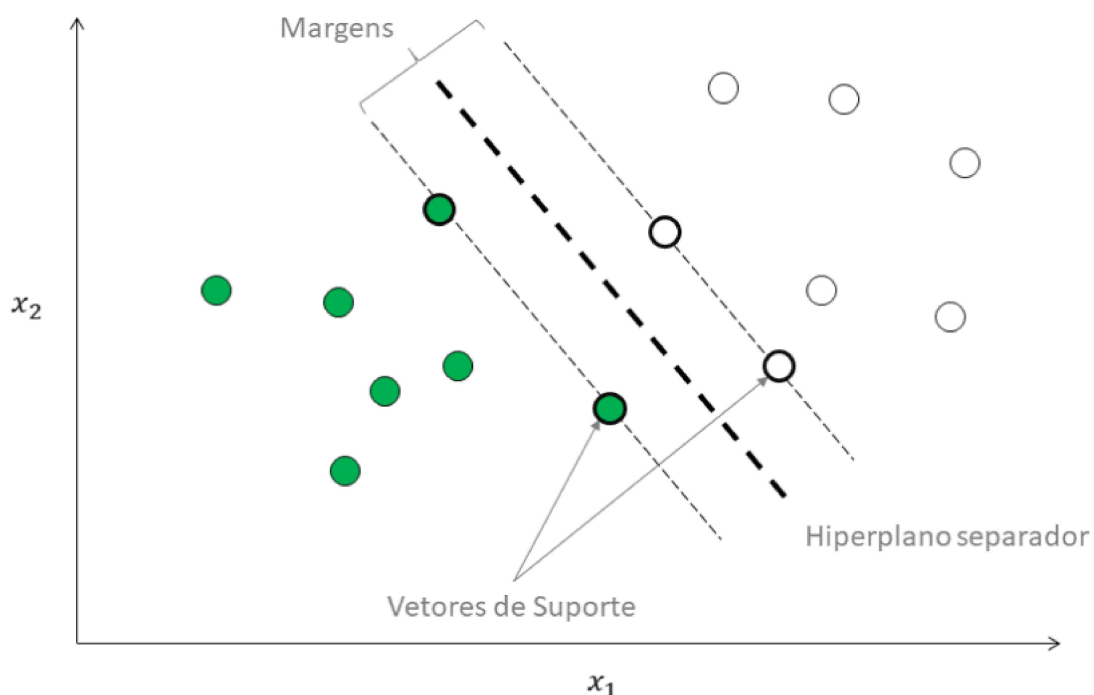
2.2 MÁQUINAS DE VETORES DE SUPORTE

As máquinas de vetores de suporte são métodos de aprendizado supervisionado amplamente aplicados em análise de dados e reconhecimento de padrões. O princípio fundamental das SVMs está na construção de soluções baseadas em um subconjunto específico dos dados de treinamento, conhecido como vetores de suporte, que desempenham um papel crucial na definição do modelo final (KOCH, 2022).

Durante o treinamento, o algoritmo SVM utiliza os vetores de suporte para determinar um hiperplano que será a base para o resultado final. Esse hiperplano é definido em um espaço de N dimensões, onde N corresponde ao número de características do conjunto de dados. Em um espaço bidimensional, o hiperplano é uma linha, enquanto em um espaço tridimensional, é um plano. Em dimensões superiores, o conceito de separação ótima continua válido, com o modelo sendo ajustado para operar nesse ambiente multidimensional (AWAD; KHANNA, 2015).

Nos problemas de classificação, o objetivo principal do SVM é encontrar o hiperplano ótimo que separa duas classes maximizando a margem de separação, ou seja, a distância entre o hiperplano e os vetores de suporte mais próximos de cada classe. Ao maximizar essa margem, o SVM promove uma melhor generalização do modelo, aumentando a probabilidade de que novos dados sejam classificados corretamente, mesmo quando não fazem parte do conjunto de treinamento (KOCH, 2022). Na Figura 1 está ilustrado um hiperplano ótimo, onde a distância entre o hiperplano e as margens está maximizada.

Figura 1 – Demonstração de um caso de classificação com hiperplano separador ótimo



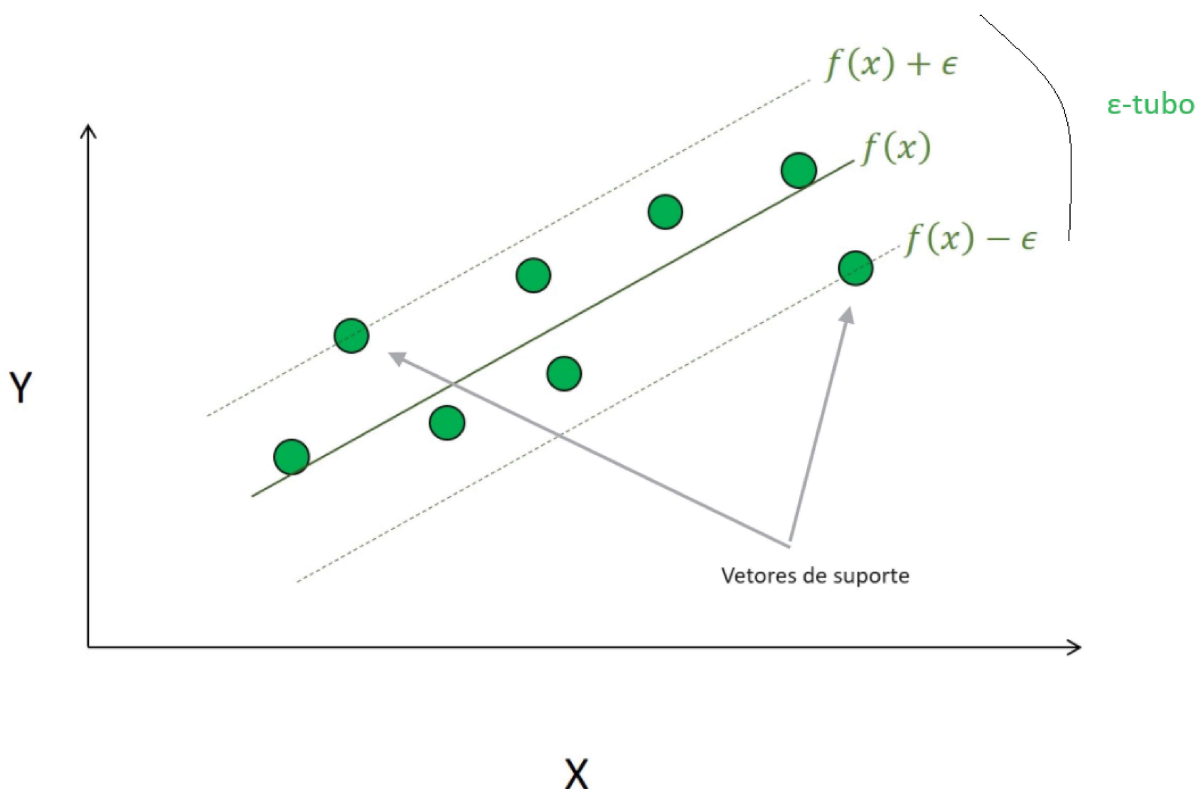
fonte: (ARA; OSPINA; MAIA, 2023).

As SVMs são especialmente eficazes para tarefas de classificação binária, mas o método também foi adaptado para outros tipos de problemas, como a regressão. Essa adaptação resultou no método conhecido como regressão com vetores de suporte, cujo objetivo é prever valores contínuos. O uso da SVR tem crescido significativamente devido à sua capacidade de lidar com grandes volumes de dados e modelar relações complexas (AWAD; KHANNA, 2015). Este trabalho concentra-se especificamente na aplicação de SVR, e seu funcionamento será detalhado a seguir.

2.3 REGRESSÃO MEDIANTE MÁQUINAS DE VETORES DE SUPORTE

A regressão com vetores de suporte é uma extensão das SVMs, projetada para lidar com problemas de regressão. Diferentemente das SVMs usadas para classificação, que buscam separar os dados em classes distintas, o SVR visa encontrar um hiperplano que aproxime os valores contínuos dos dados. Para isso, utiliza uma função de perda ϵ -insensível, que ignora erros menores que um certo valor ϵ , focando apenas nos pontos fora desse intervalo, conhecido como ϵ -tubo. Isso significa que o SVR considera apenas os pontos que estão fora de uma faixa de tolerância ao redor do hiperplano, permitindo modelar relações complexas e lidar eficientemente com grandes volumes de dados, ao mesmo tempo que reduz o risco de sobreajuste (AWAD; KHANNA, 2015). Na Figura 2 está ilustrado um caso de SVR.

Figura 2 – Demonstração de um caso de SVR



fonte: Modificado de (ARA; OSPINA; MAIA, 2023)

O objetivo do SVR é encontrar um hiperplano que melhor se ajuste aos dados, minimizando a complexidade do modelo e mantendo a precisão das previsões. A margem, no caso do SVR, é definida como a tolerância ao erro do modelo dentro do tubo insensível ϵ -tubo, ou seja, pontos de dados que

caem dentro dessa faixa não são considerados erros. A penalização para os pontos fora do tubo é controlada por um parâmetro de regularização C . Esse parâmetro equilibra a complexidade do modelo com sua capacidade de generalização: valores altos de C resultam em menos regularização e maior rigidez no ajuste aos dados, enquanto valores baixos promovem maior regularização (LORENA; CARVALHO, 2007).

Matematicamente, o SVR é formulado como um problema de otimização convexa, buscando uma função $f(x)$ que seja o mais plana possível e que tenha desvios de no máximo ϵ em relação aos alvos reais dos dados de treinamento. A planicidade é interpretada como a minimização do valor de w na função linear $f(x) = wx + b$, o que reduz a sensibilidade a pequenas variações nos dados e melhora a generalização.

Além disso, o SVR pode lidar com problemas de regressão não linear por meio de funções de *kernel*, que transformam os dados para um espaço de dimensões mais altas, onde um problema linear pode ser resolvido. Entre os *kernels* mais usados estão o linear, polinomial e função de base radial (RBF -*Radial Basis Function*). Esses kernels permitem que o SVR seja aplicado a diferentes tipos de conjuntos de dados, expandindo sua aplicabilidade (AWAD; KHANNA, 2015).

No caso de SVR baseados em RBFs, outro parâmetro importante a ser mencionado é o γ . Ele pode ser interpretado como o inverso do raio de influência das amostras selecionadas pelo modelo como vetores de suporte. Dessa forma, um valor baixo de γ significa que a influência de um único vetor de suporte tem um longo alcance, afetando uma região maior do espaço de características. Por outro lado, um valor alto de γ indica que a influência é limitada a uma região próxima ao exemplo de treinamento (Scikit-learn Developers, 2024).

3 CONFIGURAÇÕES DA SIMULAÇÃO E DO SVR

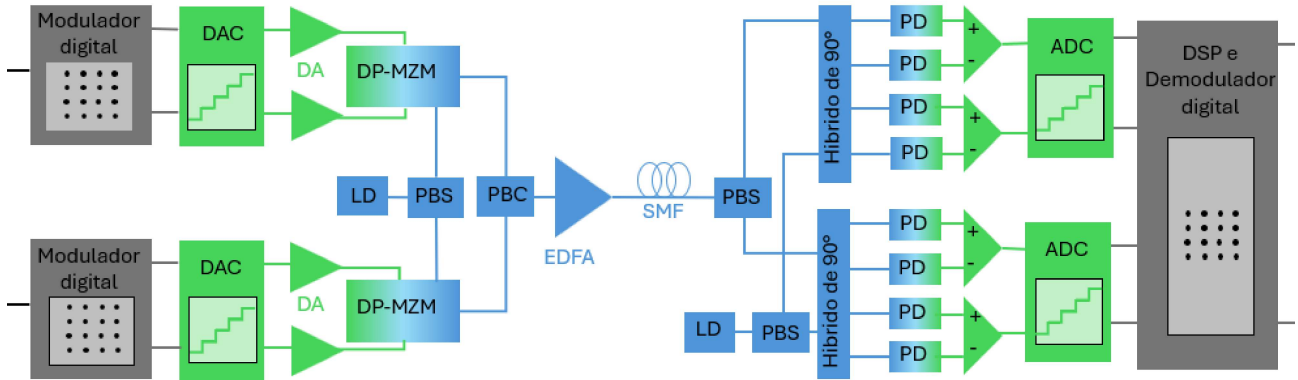
3.1 CONFIGURAÇÕES DA SIMULAÇÃO

A simulação do sistema de comunicação óptica coerente digital foi realizada utilizando o software comercial *VPI Transmission Maker*, amplamente reconhecido por sua capacidade de modelar efeitos lineares e não lineares com elevado grau de realismo. Este estudo concentrou-se em um sistema baseado na modulação 16-QAM com dupla polarização e abordou a mitigação de distorções não lineares por meio do método de regressão com vetores de suporte, implementada no ambiente *Jupyter Notebook* com a linguagem de programação *Python* e a biblioteca *scikit-learn*.

O sistema simulado consistiu em um enlace de fibra monomodo (SMF - *Single Mode Fiber*) com comprimento de 140 km. No transmissor, sequências pseudoaleatórias de bits (PRBS - *Pseudo Random Binary Sequence*), com taxas de até 112 Gb/s, foram divididas em duas subsequências e mapeadas em constelações 16-QAM. Após a superamostragem, os sinais foram filtrados por filtros (RCF - *Raised-cosine Filter*) com fator de roll-off de 20% e modulados por dois moduladores Mach-Zehnder duplo-paralelo. Esses moduladores operaram em duas polarizações ortogonais provenientes de lasers de onda contínua com largura de linha entre 100 e 200 kHz. Os sinais modulados foram combinados por um combinador de polarização, e amplificados por um amplificador óptico dopado com érbio (EDFA - *Erbium-doped Fiber Amplifier*) com figura de ruído de 4 dB, ajustando a potência de transmissão entre 1 e 12 dBm.

No receptor, o sinal foi detectado utilizando um receptor coerente com diversidade de polarização e fase. As componentes em fase e em quadratura foram digitalizadas e processadas por um DSP (*Digital Signal Processor*). O processamento incluiu a ortogonalização, para corrigir desbalanceamentos no híbrido óptico, a compensação de dispersão cromática no domínio da frequência e a equalização dinâmica utilizando o algoritmo de múltiplos módulos (MMA - *Multimodulus Algorithm*), responsável por mitigar a dispersão de modo de polarização e o crosstalk linear residual. Adicionalmente, a compensação de ruído de fase foi realizada por meio de técnicas de busca cega. O diagrama de blocos do sistema óptico simulado está apresentado na Figura 3.

Figura 3 – Diagrama de blocos da configuração da simulação. DAC: Conversor digital-analógico (*Digital-to-Analog Converter*), DA: Amplificador de driver (*Driver Amplifier*), LD: Laser de diodo (*Diode Laser*), PBS: Divisor de feixe de polarização (*Polarization Beam Splitter*), DP-MZM: Modulador Mach-Zehnder duplo-paralelo (*Dual-Parallel Mach-Zehnder Modulator*), PBC: Combinador de feixe de polarização (*Polarization Beam Combiner*), EDFA: Amplificador de fibra dopada com érbio (*Erbium-Doped Fiber Amplifier*), SMF: Fibra monomodo (*Single-Mode Fiber*), PD: Fotodetector (*photodetector*) e ADC: Conversor analógico-digital (*Analog-to-Digital Converter*).



fonte: Produção do próprio autor.

3.2 CONFIGURAÇÃO DO SVR

Para avaliar o desempenho do SVR na mitigação das distorções não lineares, e comparar de forma justa com o desempenho do SVC, foram considerados dois modos de processamento: independente, com polarizações tratadas separadamente, e conjunto, abordando ambas simultaneamente. Essa abordagem possibilitou uma análise comparativa entre as configurações, identificando o cenário de melhor desempenho. Adicionalmente, a otimização dos parâmetros C e γ foi realizada para as abordagens de polarização única e dupla polarização, visando obter a configuração mais eficiente para compensar os efeitos não lineares.

O *kernel* RBF foi selecionado por sua capacidade de modelar sistemas não lineares de forma flexível, sendo particularmente eficaz para dados que não podem ser separados por fronteiras lineares. Para refinar os resultados, os parâmetros foram ajustados, com C variando entre 0.2, 1 e 10, e γ entre 0.1, 0.5 e 1.

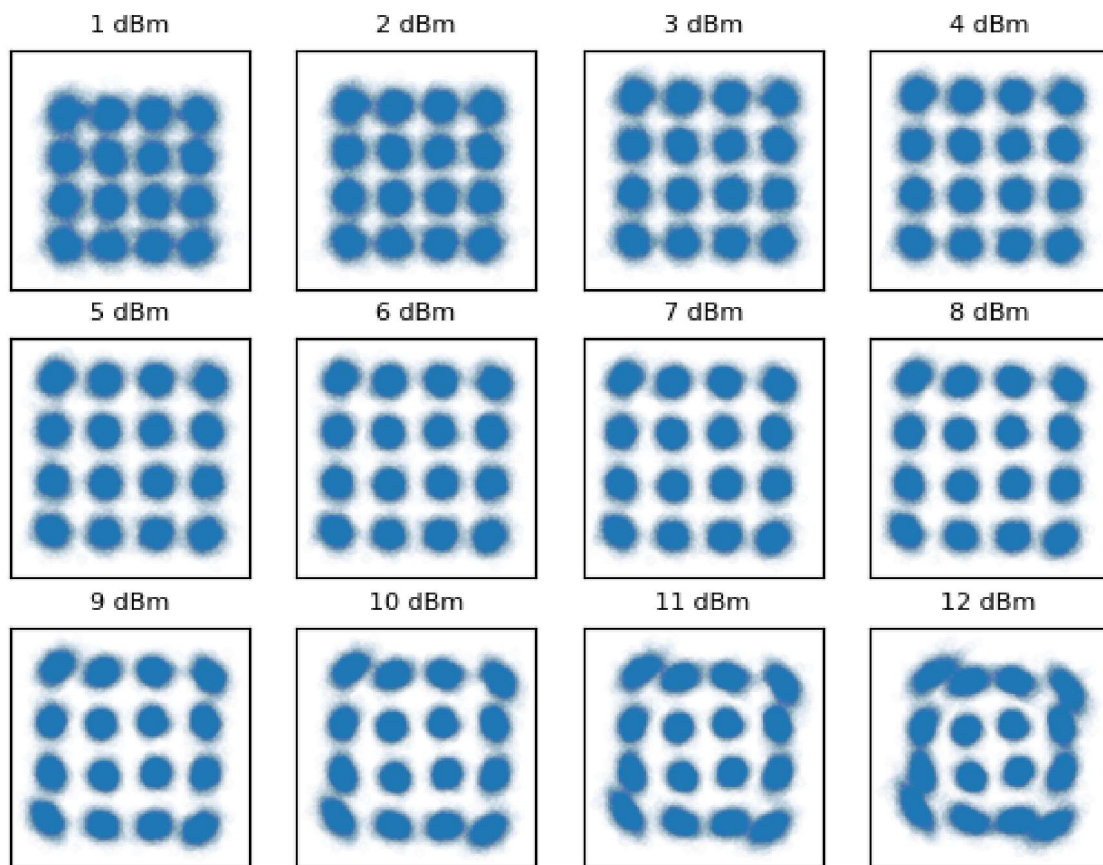
A avaliação de desempenho foi baseada na taxa de erro de bit, obtida por meio de equalização e contagem direta de erros utilizando o método de detecção por máxima verossimilhança (ML - *Maximum Likelihood*). A análise comparativa incluiu o desempenho do SVR, do SVC e do método de detecção por ML aplicado sem a utilização de algoritmos, permitindo uma análise abrangente da eficácia de cada abordagem na mitigação dos erros induzidos por não linearidades.

4 RESULTADOS

4.1 ANÁLISE DE DESEMPENHO

Com base nos dados obtidos por meio das simulações realizadas no software *VPI Transmission Maker*, foram gerados diagramas de constelação para todas as potências de lançamento avaliadas neste estudo. Na Figura 4, é possível observar a cada gráfico o aumento das distorções no diagrama de constelação, assim fica evidente o impacto dos efeitos não lineares em sistemas coerentes digitais, demonstrando como esses efeitos se intensificam progressivamente com o aumento da potência de transmissão.

Figura 4 – Diagrama de constelação para potências de transmissão de 1 a 12dBm



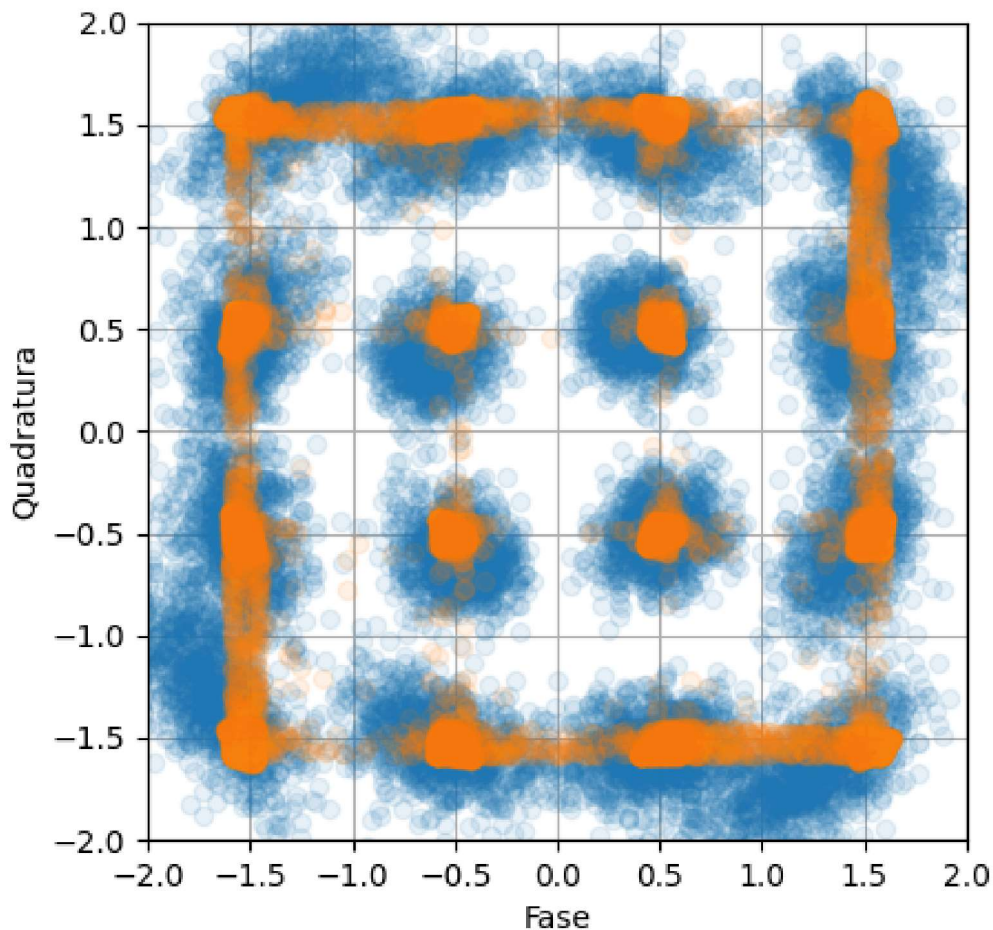
fonte: Produção do próprio autor.

Para avaliar o desempenho do algoritmo SVR na mitigação dos efeitos não lineares, este método foi aplicado às constelações geradas para diferentes potências ópticas de transmissão em dois cenários diferentes, o primeiro considera o processamento das polarizações de forma individual (SP - *Single Polarization*) e o segundo considera o processamento das duas polarizações de forma conjunta (DP

- *Dual polarization*). Na Figura 5 e na Figura 6, são apresentados os diagramas de constelação contendo os resultados obtidos para uma potência de 12 dBm, considerando o processamento SP e DP respectivamente. Estes gráficos permitem visualizar o impacto da técnica na correção das distorções causadas pelas não linearidades do sistema.

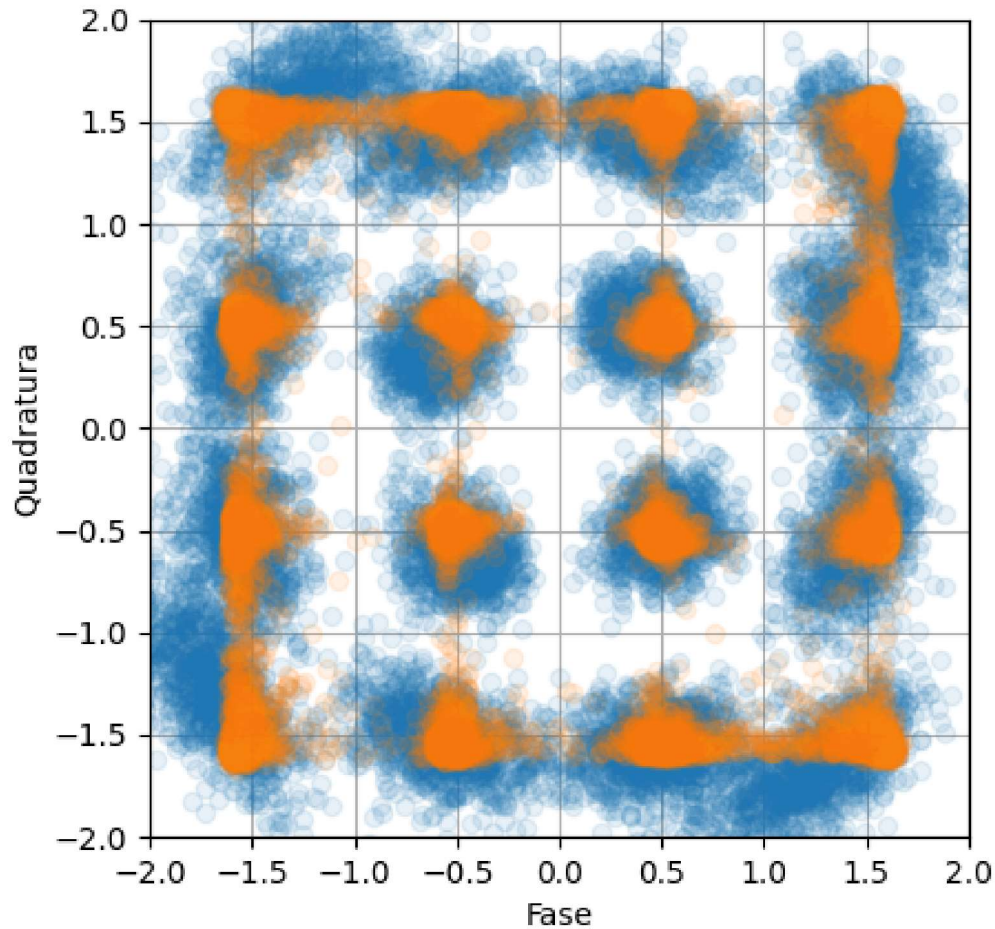
Nas figuras Figura 5 e Figura 6, os elementos em azul representam o diagrama de constelação antes da aplicação do algoritmo SVR, enquanto os elementos em laranja correspondem ao diagrama após sua aplicação. Em ambos os casos, observa-se que os pontos laranja estão mais concentrados e centralizados em relação aos pontos azuis, evidenciando uma constelação mais definida e próxima dos valores ideais, o que reflete a eficácia na correção dos efeitos não lineares. No entanto, ao comparar os dois gráficos, percebe-se que o diagrama de dupla polarização apresenta um espalhamento discretamente menor, especialmente nas extremidades, em relação ao diagrama de polarização única, ficando, portanto, ligeiramente mais próximo da constelação ideal.

Figura 5 – Diagrama de Constelação para SP e potência de transmissão 12 dBm



fonte: Produção do próprio autor.

Figura 6 – Diagrama de Constelação para DP e potência de transmissão 12 dBm



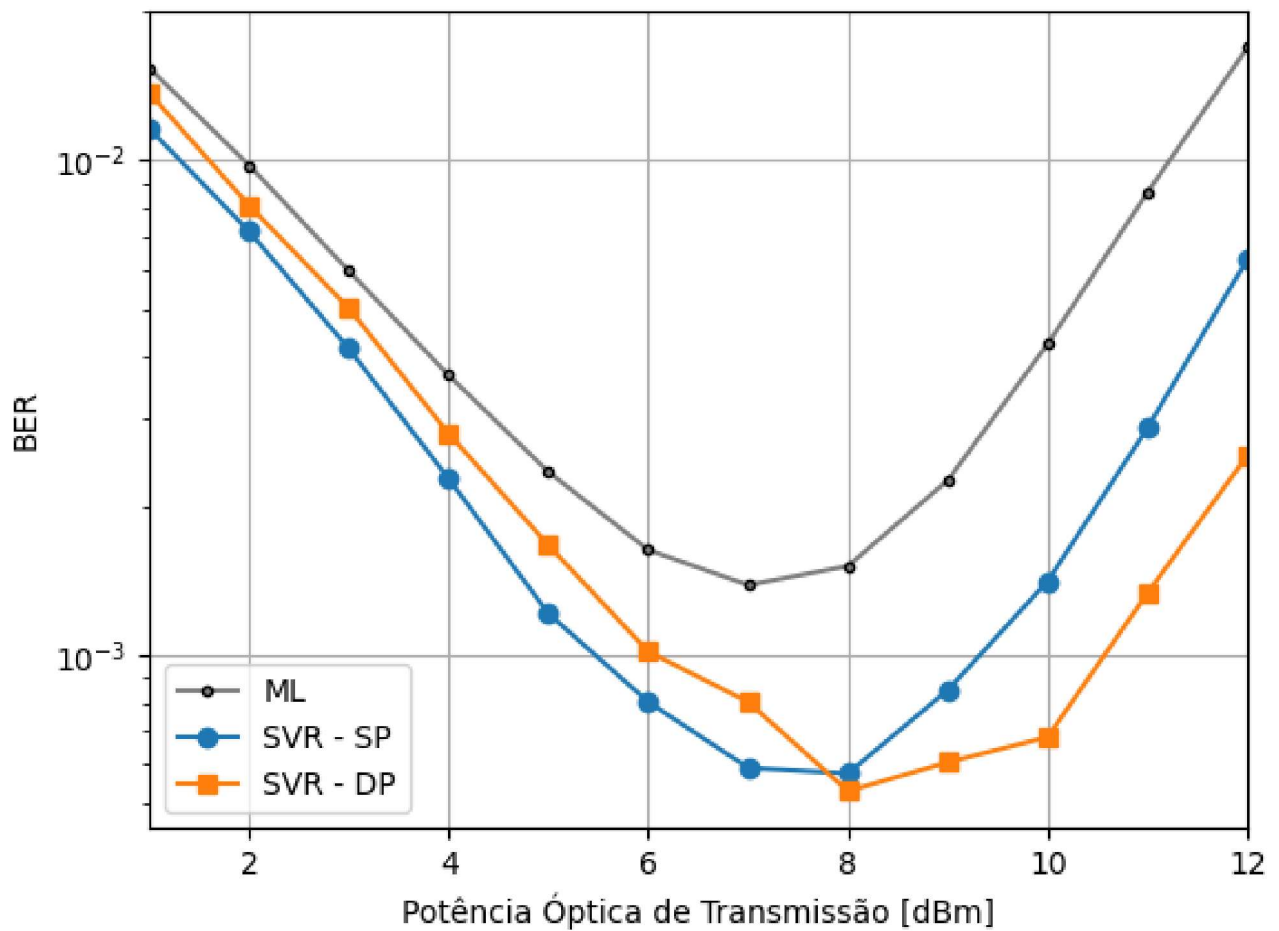
fonte: Produção do próprio autor.

A Figura 7 exibe os resultados da aplicação do algoritmo SVR utilizando processamento SP e DP, utilizando os parâmetros $\gamma = 1$ e $C = 1$, em comparação a aplicação do método de detecção por máxima verossimilhança (ML - *Maximum likelihood*), sem o uso do algoritmo SVR.

Os dados indicam que, em ambos os cenários avaliados, o uso do SVR promove uma redução na BER para todos os níveis de potência óptica de transmissão analisados. Essa melhoria é particularmente mais expressiva em potências mais elevadas, evidenciando a capacidade do SVR em compensar parcialmente os efeitos das distorções não lineares.

Além disso, é possível observar na Figura 7 uma diferença de desempenho entre as abordagens SP e DP. Para potências de transmissão superiores a 8 dBm, a dupla polarização apresenta uma menor BER em comparação com a polarização única, indicando maior eficácia na mitigação de efeitos não lineares para este sistema. Contudo, para potências mais baixas, que são menos afetadas por distorções não lineares, a polarização simples demonstrou melhor desempenho.

Figura 7 – BER em função da potência óptica de transmissão considerando processamento DP e SP.



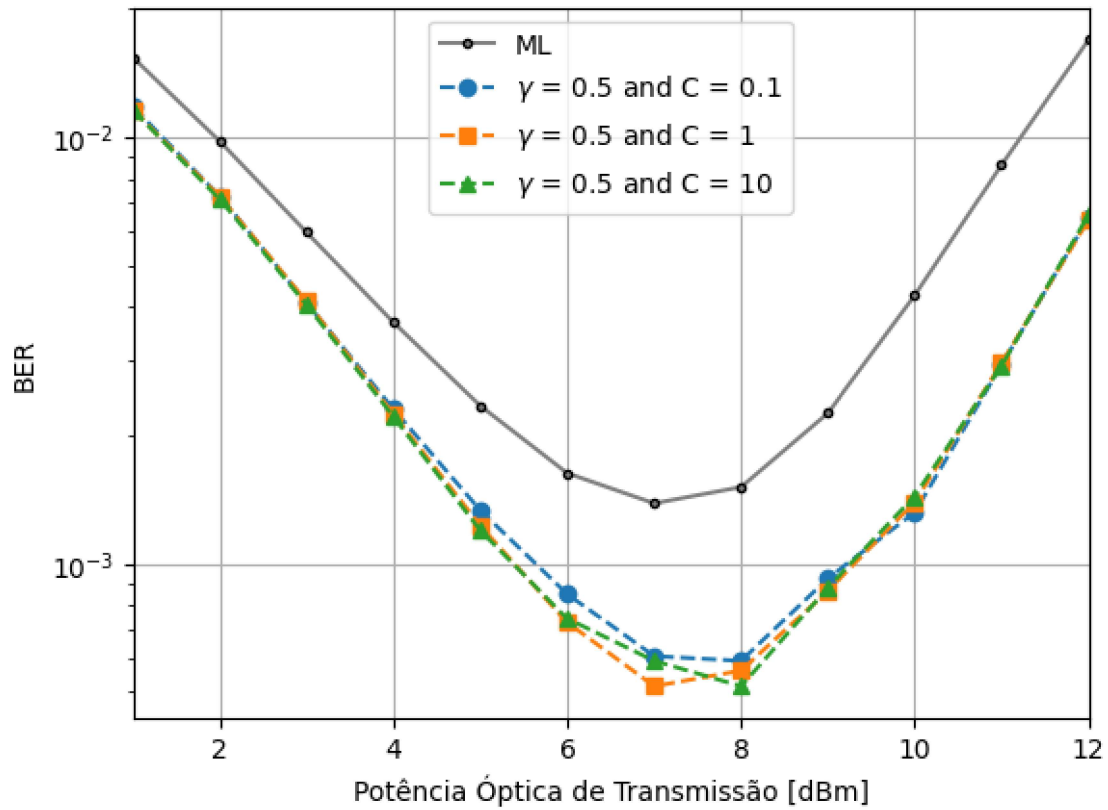
fonte: Produção do próprio autor.

A seguir, são apresentados os resultados da otimização dos parâmetros C , visando identificar o cenário de melhor desempenho do SVR para as abordagens SP e DP.

Na Figura 8, observa-se que, ao variar o coeficiente de regularização C no intervalo de 0.1 a 10 para a polarização simples, as curvas de BER permanecem praticamente inalteradas. É perceptível que, mesmo com um valor baixo do coeficiente de regularização como $C = 0.1$, o desempenho do SVR com a polarização simples mantém-se satisfatório quando comparado ao método de ML.

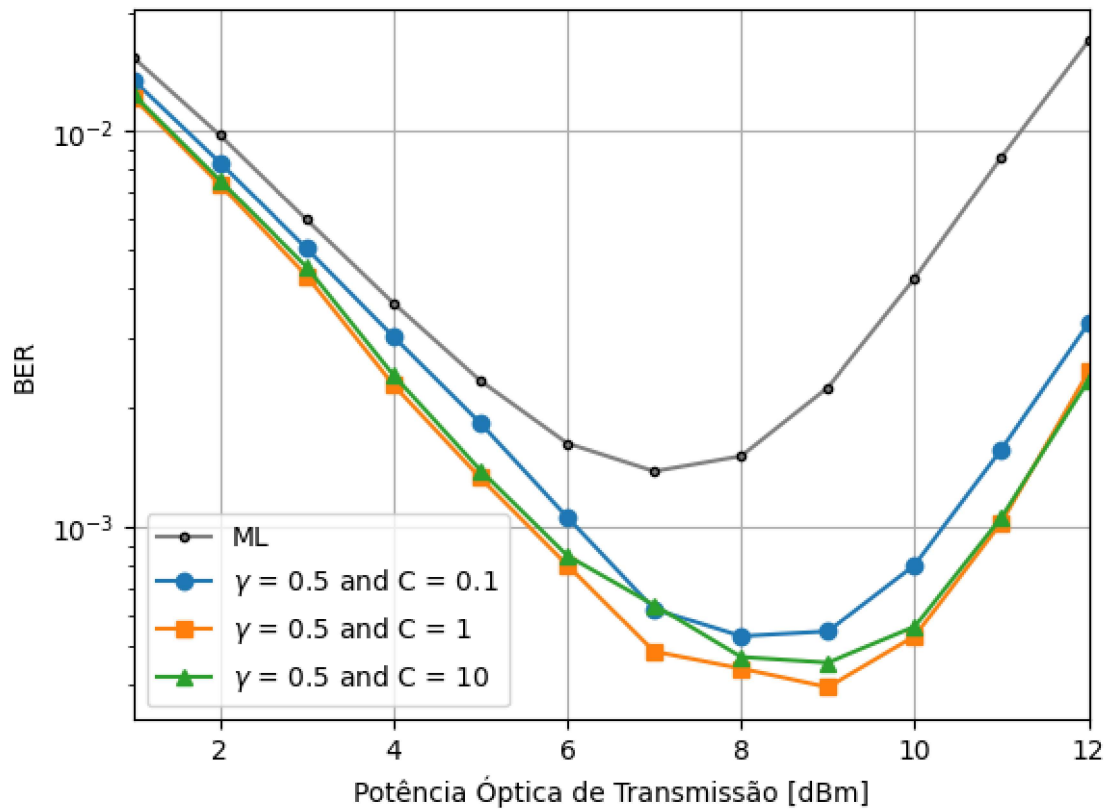
Já na Figura 9, são apresentadas as curvas de BER obtidas para DP quando o coeficiente de regularização C é variado. Nesse caso, a diferença entre a curva correspondente a $C = 1$ e a curva correspondente a $C = 0.1$ é mais perceptível, porém, essa diferença não é muito expressiva em termos de desempenho geral. Isso indica que, para o caso analisado neste trabalho, a variação no valor de C não exerce um impacto significativo no desempenho do SVR, que continua apresentando resultados superiores à ML para todos os níveis de potência avaliados.

Figura 8 – BER em função da potência óptica considerando SP e diferentes valores do parâmetro C.



fonte: Produção do próprio autor.

Figura 9 – BER em função da potência óptica considerando DP e diferentes valores do parâmetro C.



fonte: Produção do próprio autor.

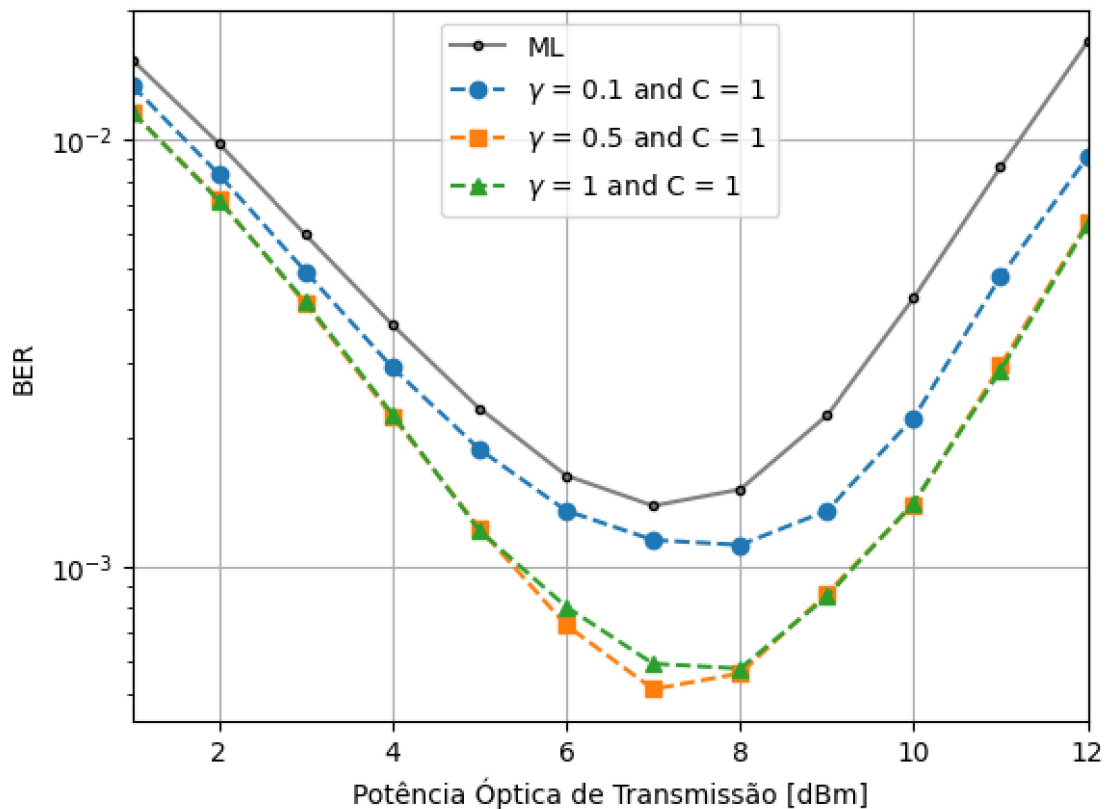
Além do parâmetro C , também foi realizada a otimização do parâmetro γ , tanto para a abordagem SP quanto para a abordagem DP. Os resultados estão apresentados a seguir.

Com a variação do parâmetro γ , observa-se uma diferença mais acentuada nas curvas de BER, indicando que, no caso analisado neste trabalho, esse parâmetro exerce um impacto mais significativo no desempenho do SVR, especialmente em níveis mais elevados de potência.

Tanto para SP quanto para DP, a curva correspondente a $\gamma = 0.1$ apresenta um desempenho inferior, mais próximo ao do método ML. Isso ocorre porque valores muito baixos de γ acarretam em problemas de subajuste. Em ambos os casos, um valor intermediário de γ , como $\gamma = 0.5$, demonstra o melhor desempenho nos resultados.

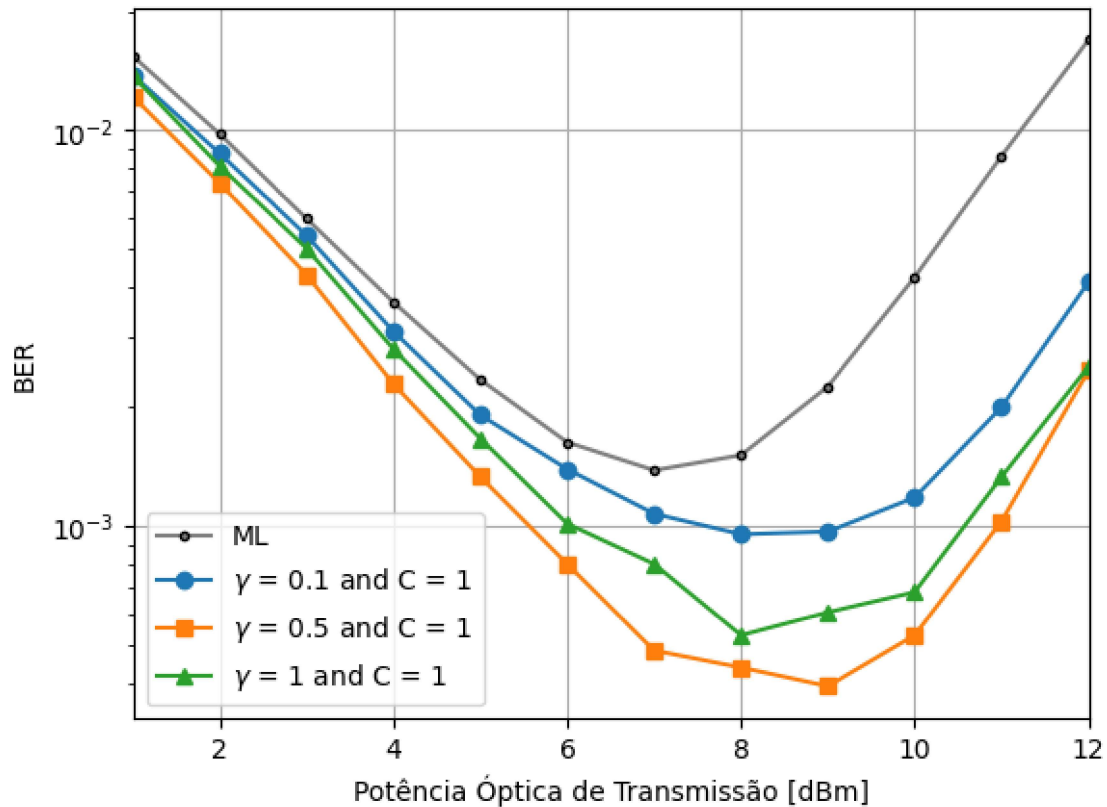
Na abordagem SP, quase não se observa diferença entre as curvas de $\gamma = 0.5$ e $\gamma = 1$. Por outro lado, na abordagem DP, a curva de $\gamma = 1$ apresenta uma degradação do desempenho em relação à curva $\gamma = 0.5$, o que pode ser explicado pelo fato de que valores muito altos de γ geram erro por sobreajuste. Os efeitos de sobreajuste são mais evidentes quando ambas as polarizações são processadas simultaneamente, uma vez que a dimensionalidade do modelo aumenta em comparação ao processamento independente de cada polarização.

Figura 10 – BER em função da potência óptica considerando SP e diferentes valores de γ .



fonte: Produção do próprio autor.

Figura 11 – BER em função da potência óptica considerando DP e diferentes valores de γ .



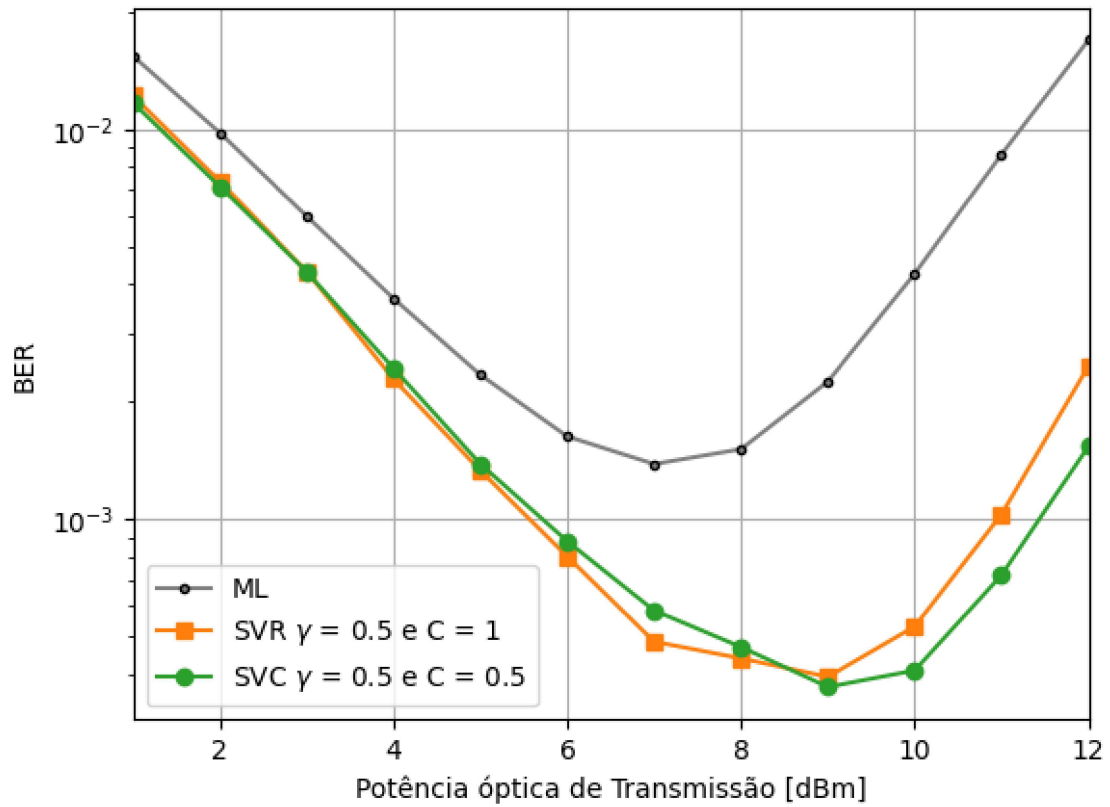
fonte: Produção do próprio autor.

4.2 COMPARAÇÃO COM UM SISTEMA SVC

Após a otimização dos parâmetros da SVR, podemos realizar a comparação de seus resultados com os resultados da SVC apresentados no artigo (ALDAYA et al., 2023).

Na Figura 12, observa-se que independentemente da abordagem adotada, o SVM apresenta uma diminuição na BER em toda a faixa de potências ópticas consideradas. Isso evidencia que o SVM é capaz de mitigar não apenas os efeitos de distorção não linear, mas também alguns efeitos lineares residuais. À medida que a potência óptica aumenta, a distorção não linear da fibra se torna mais relevante, ampliando a diferença entre as curvas de BER para ML e SVM. Podemos observar também que até o valor de 9 dBm, a diferença de desempenho entre SVR e SVC não é muito significativa. A partir de 10 dBm, nota-se que a SVC apresenta uma sutil vantagem em relação à SVR, demonstrando que, para a mitigação de efeitos não lineares deste sistema, que se tornam mais presentes com maiores potências ópticas, SVC oferece um desempenho ligeiramente superior.

Figura 12 – Comparação SVR, SVC e ML



fonte: Produção do próprio autor.

5 CONCLUSÃO

Este estudo investigou os impactos de efeitos não lineares em um sistema óptico coerente digital de canal único, considerando um enlace de 140 km e modulação no formato dupla polarização 16QAM, utilizando o software *VPI Transmission Maker* para simulação. Também foram analisados os desempenhos de algoritmos de máquinas de vetores de suporte, nas abordagens de classificação e regressão, para mitigação desses efeitos.

Foi implementado um algoritmo de regressão por vetores de suporte com o objetivo de compensar as adversidades causadas pelas não linearidades, sendo seu desempenho comparado ao algoritmo de classificação por vetores de suporte e ao método convencional de detecção por máxima verossimilhança.

A implementação do algoritmo resultou em reduções na taxa de erro de bits em sistemas limitados por não linearidades, em comparação com o método de detecção ML convencional. Entretanto, em cenários onde a potência de transmissão é menor e assim o sistema acaba sendo menos afetado por não linearidades, os algoritmos SVM apresentaram desempenho similar ao da detecção ML, o que limita sua viabilidade devido ao alto custo computacional.

Os resultados foram avaliados em sistemas com processamento de polarização única e dupla, com ambos apresentando melhorias na BER. No entanto, as diferenças de desempenho entre as duas abordagens foram pequenas, com os sistemas de dupla polarização mostrando uma sutil vantagem em cenários de potências de transmissão mais elevadas.

Já a comparação entre os sistemas otimizados de SVR e SVC, ambos aplicados a cenários de dupla polarização, revelou que o SVC apresentou uma BER ligeiramente inferior para potências de transmissão acima de 9 dBm. Esse comportamento sugere um desempenho sutilmente superior do SVC na mitigação de efeitos não lineares. No entanto, a diferença observada entre as curvas de desempenho é pouco expressiva, e neste caso, o SVC não apresenta uma vantagem significativa sobre o SVR.

Assim, para o sistema em questão, o desempenho do SVR é similar ao do SVC. Dessa forma, demonstramos que o uso do SVR não proporciona uma melhoria significativa, o que não estava claramente evidenciado na literatura.

Este estudo considerou sistemas de comunicação óptica de canal único, mas pode ser ampliado futuramente para sistemas WDM.

REFERÊNCIAS

- AGRAWAL, G. P. **Nonlinear Fiber Optics**. Burlington, MA: Academic Press, 2007.
- AGRAWAL, G. P. **Fiber-optic Communication Systems**. New York, USA: John Wiley & Sons, 2021.
- ALDAYA, I. et al. Impairment mitigation in dual-polarization single-span optical digital coherent systems using support vector classifiers. **XLI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2023)**, São José dos Campos, SP, October 2023.
- ALPAYDIN, E. **Introduction to Machine Learning**. 3rd. ed. Cambridge, Massachusetts, London, England: The MIT Press, 2014. ©2014 Massachusetts Institute of Technology.
- ARA, A.; OSPINA, R.; MAIA, M. **Modelos de Vetores de Suporte: Uma Introdução ao Aprendizado Estatístico de Máquina**. 2023. 67ª RBras e 20º SEAGRO, Londrina-PR. Disponível em: <http://leg.ufpr.br/~ara/teach/svm/01_introducao.html>.
- AWAD, M.; KHANNA, R. **Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers**. [S.l.]: Springer Nature, 2015.
- BARBOSA, F. A. **Avaliação da Interação entre a Formatação Probabilística de Constelações e Etapas de Processamento Digital de Sinais para Sistemas Ópticos Coerentes Digitais**. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, Campinas, 2021.
- BENINCA, M. O. L. **Estudo sobre Amplificadores Híbridos Raman-EDFA**. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória, 2012. Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico.
- BORGES, L. N. **Uso de algoritmos de clusterização para mitigação de impedimentos em sistemas ópticos coerentes digitais DP-16QAM de alta taxa de transmissão de um único span**. Dissertação (Mestrado) — Universidade Estadual Paulista Júlio de Mesquita Filho - Campus de São João da Boa Vista, São João da Boa Vista, 2021.
- DINIZ, J. C. M. **Estimador de Desvio de Frequência para Receptores Ópticos Coerentes Digitais**. Dissertação (Dissertação de Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, 2013.
- KEISER, G. **Optical Fiber Communications**. 2nd. ed. New York, USA: McGraw-Hill, 1991.
- KOCH, R. **Application of Machine Learning in Optical Communications**. Tese (Dissertation) — Christian-Albrechts-Universität zu Kiel, 2022.
- LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Uma introdução às support vector machines. **Publicação do Centro de Matemática, Computação e Cognição, Universidade Federal do ABC e Departamento de Ciências de Computação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo**, Santo André, SP e São Carlos, SP, 2007.
- MAATEN, L. van der; POSTMA, E.; HERIK, J. van den. **Dimensionality Reduction: A Comparative Review**. [S.l.], 2009.
- MIRANDA, T. D. L. **Análise de efeitos não-lineares em fibras ópticas monomodo**. Dissertação (Mestrado) — Instituto Nacional de Telecomunicações (INATEL), Santa Rita do Sapucaí, 2011.

NUNES, F. D. **Sistemas de Comunicação Óptica – Projetando Enlaces Ópticos**. Pernambuco, BR: Universidade de Pernambuco, 2001.

PINTO, A. N. **Análise e Optimização de Sistemas de Comunicação Ópticos Baseados em Solitões**. Tese (Doutorado) — Universidade de Aveiro, Aveiro, Portugal, 1999.

ROUDAS, I. Wdm systems and networks. **Optical Networks**, Springer, New York, 2011.

Scikit-learn Developers. **SVM with RBF Kernel: Tuning Parameters Example**. 2024.
<https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html>. Acesso em: 20 nov. 2024.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning: From Theory to Algorithms**. [S.l.]: Cambridge University Press, 2014.

YE, X. **Applications of Artificial Intelligence to Control and Analyze the Performance of Fiber-Optic Transmission Systems**. Tese (PhD Thesis) — Institut Polytechnique de Paris, 2023.

APÊNDICE A – CÓDIGO EM PYTHON

A.1 COMPENSAÇÃO DE EFEITOS NÃO LINEARES UTILIZANDO SVR

```

breakatwhitespace
    # Import libraries
    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    # Configure the environment
    %matplotlib qt
    def MLdetection_16QAM(X, lev_sep=1):
        # Building the decision region limits
        lims = (np.arange(4)-1.5)*lev_sep
        m, n = np.shape(X)
        int_label = np.zeros(m)
        for aux1 in range(n):
            for aux2 in range(m):
                int_label[aux2] = int(int_label[aux2]+np.round
((X[aux2,aux1]-lims[0]))*4**aux1)
        return int_label

    def plot_by_label(X, labels, M=16):
        plt.figure()
        for aux in range(M):
            ind = np.where(labels==aux)
            plt.plot(X[ind[0],0],X[ind[0],1],'.')
        plt.legend(['0','1','2','3','4','5','6','7','8','9','10',
', '11','12','13','14','15'], bbox_to_anchor=(1, 1))
        plt.axis('square')

    def constellation_hist(X):
        plt.figure()
        plt.hist2d(X[:,0],X[:,1])

    def sort_clusters(labels_clusters, labels_ideal, M=16):

        labels_new = np.zeros(len(labels_clusters), dtype=int)

```

```

    for aux in range(M):
        ind = np.where(labels_ideal==aux)
        lab = labels_clusters[ind]
        new_label = int(np.round(np.mean(lab)))
        ind_1 = np.where(labels_clusters==new_label)
        labels_new[ind_1] = aux
    return labels_new

def compare_labels(labels_dist, labels_ideal):
    error_counter = 0
    for aux in range(len(labels_dist)):
        if labels_dist[aux] != labels_ideal[aux]:
            error_counter += 1
    return error_counter

1. Load and plot data
# Close all figures
plt.close('all')

# Number of symbols to process
Nsym = 50000

# Load ideal data
folder_name = 'Dados_DP-16QAM_100Gbps/Dados_140km_14e9Bd'
file_name = 'DP_IdealConstellationDiagram_0dbm_140km_14e9Bd'
.csv'

complete_name = folder_name+'/'+file_name
print('Loading (ideal data):', complete_name)
dataframe = pd.read_csv(complete_name, sep=',', header=5)
data_ideal = dataframe.to_numpy()
m, n = np.shape(data_ideal)

# Reduce the shape of ideal data
#print('\nThe shape of original data_ideal is:', np.shape(
data_ideal))
data_ideal = data_ideal[:Nsym, :]
#print('The shape of reduced data_ideal is:', np.shape(
data_ideal))

# Convert to integer labels (considering each polarization
independently)

```

```

labels_X = MLdetection_16QAM(data_ideal[:,0:2])
labels_Y = MLdetection_16QAM(data_ideal[:,2:4])
labels_XY = MLdetection_16QAM(data_ideal[:,:])

# Plot each label in a different colour
plot_by_label(data_ideal[:,0:2],labels_X)
print(data_ideal[0,:])
# Load distorted data for 140 km
folder_name = 'Dados_DP-16QAM_100Gbps/Dados_140km_14e9Bd'
file_name_list = ['#'
DP_RealConstellationDiagram_0dbm_100km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_1dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_2dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_3dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_4dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_5dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_6dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_7dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_8dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_9dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_10dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_11dbm_140km_14e9Bd.csv',
                    ,
DP_RealConstellationDiagram_12dbm_140km_14e9Bd.csv']

# Generation of the storing tensor
k = len(file_name_list)
data = np.zeros((k,Nsym,n))
# Iterate over the elements of the list

```

```

for counter, file_name in enumerate(file_name_list):
    complete_name = folder_name+'/'+file_name
    print('Loading (distorted data):',complete_name)
    dataframe = pd.read_csv(complete_name,sep=',', header
=5)

    data_aux = dataframe.to_numpy()
    data[counter,:,:] = data_aux[:Nsym,:]
Graphical representation

# Plot distorted data
points_to_plot = 10000
plt.figure()
for ind in range(12):
    ax = plt.subplot(3,4,ind+1)
    plt.plot(data[ind,:points_to_plot,0],data[ind,:
points_to_plot,1],'.',alpha=0.01, color = 'C0')
    plt.plot(data[ind,:points_to_plot,2],data[ind,:
points_to_plot,3],'.',alpha=0.01, color = 'C0')
    plt.axis('square')
    plt.title(f"{ind+1} dBm")
    ax.set_xticks([])
    ax.set_yticks([])

BER estimation by detecting using ML and comparing with the
ideal labels

BER_array = np.zeros(k)

for ind in range(k):
    print('Processando',ind,'de',k-1)
    data_dist = data[ind,:,:]
    labels_X_dist = MLdetection_16QAM(data_dist[:,0:2]/2)
    labels_Y_dist = MLdetection_16QAM(data_dist[:,2:4]/2)
    labels_XY_dist = MLdetection_16QAM(data_dist[:,,:]/2)

    BER_X = sum(labels_X!=labels_X_dist)/len(labels_XY)/4
    BER_Y = sum(labels_Y!=labels_Y_dist)/len(labels_XY)/4
    BER_XY = sum(labels_XY!=labels_XY_dist)/len(labels_XY)

/8

BER_array[ind] = (BER_X+BER_Y)/2

```

```

    print('\tBER is:',BER_XY)
plt.figure()
P_array = np.arange(1,k+1)
plt.semilogy(P_array,BER_array)
plt.xlabel('Potencia de Transmissao [dBm]')
plt.ylabel('BER')
plt.grid('on')
plt.title('BER para Maxima Verossimilhanca')
Regression using Support Vector Regression of each
polarization for a given input power level

from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

#Polarizacao X
# We select a mid-power constellation
ind = 11

# Data and labels
DataX = data[ind,:,2:]          # We select X polarization
LabelsX = labels_X             # We select labels of X
polarization

# Rename to the meet the variables names of sklearn
X = DataX[:,:]                # Input values
Y = data_ideal[:,0:2]         # Output values

scaler = StandardScaler()
scaler.fit(X)
scaler.transform(X)
scaler.fit(Y)
scaler.transform(Y)

# Split data into train and test
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.33, random_state=42)

# Perform the regression for X polarization

```

```

    regr_inphase = svm.SVR(kernel='rbf', gamma=0.5, C=1).fit(
X_train, Y_train[:,0])
    regr_quadrature = svm.SVR(kernel='rbf', gamma=0.5, C=1).fit
(X_train, Y_train[:,1])

    # Predict the in-phase and quadrature components
    y_hat_inphase = regr_inphase.predict(X_test)
    y_hat_quadrature = regr_quadrature.predict(X_test)

    # Representation of the constellation
    plt.figure()
    plt.plot(y_hat_inphase,y_hat_quadrature,'o',alpha=0.1)
    plt.title('Diagrama de Constelacao para a polarizacao X')

    # We combine the in-phase and quadrature components
    data_hat_x = np.vstack((y_hat_inphase,y_hat_quadrature)).
transpose()
    data_ideal_x = Y_test

    labels_hat_x = MLdetection_16QAM(data_hat_x)
    labels_ideal_x = MLdetection_16QAM(data_ideal_x)

    BER = sum(labels_hat_x!=labels_ideal_x)/len(labels_hat_x)/4
    print('A BER obtida para a polarizacao X e:', BER)
    # Representation of the constellation
    plt.figure()
    plt.plot(X_test[:,0]/2,X_test[:,1]/2,'o',alpha=0.1)
    plt.plot(y_hat_inphase,y_hat_quadrature,'o',alpha=0.1)
    plt.axis('square')
    plt.ylim([-2,2])
    plt.xlim([-2,2])
    plt.grid('on')
    plt.xlabel('In-phase component')
    plt.ylabel('Quadrature component')
    #Polarizacao Y
    # We select a mid-power constellation
    ind = 11

    # Data and labels for that LOP level
    DataY = data[ind, :, 2:]          # We select Y polarization

```

```

    LabelsY = labels_Y          # We select labels of Y
    polarization

    # Rename to the meet the variables names of sklearn
    X = DataY[:, :]            # Input values
    Y = data_ideal[:, 2:4]      # Output values

    scaler = StandardScaler()
    scaler.fit(X)
    scaler.transform(X)
    scaler.fit(Y)
    scaler.transform(Y)

    # Split data into train and test
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    test_size=0.33, random_state=42)

    # Perform the regression for Y polarization
    regr_inphase = svm.SVR(kernel='rbf', gamma=0.5, C=1).fit(
    X_train, Y_train[:, 0])
    regr_quadrature = svm.SVR(kernel='rbf', gamma=0.5, C=1).fit
    (X_train, Y_train[:, 1])

    # Predict the in-phase and quadrature components
    y_hat_inphase = regr_inphase.predict(X_test)
    y_hat_quadrature = regr_quadrature.predict(X_test)

    # Representation of the constellation
    plt.figure()
    plt.plot(y_hat_inphase, y_hat_quadrature, 'o', alpha=0.1)
    plt.title('Diagrama de Constelacao para a polarizacao X')

    # We combine the in-phase and quadrature components
    data_hat_y = np.vstack((y_hat_inphase, y_hat_quadrature)).
    transpose()
    data_ideal_y = Y_test

    labels_hat_y = MLdetection_16QAM(data_hat_y)
    labels_ideal_y = MLdetection_16QAM(data_ideal_y)

```

```

BER = sum(labels_hat_y!=labels_ideal_y)/len(labels_hat_y)/4
print('A BER obtida para a polarizacao Y e:', BER)
# Representation of the constellation
plt.figure()
plt.plot(X_test[:,0]/2,X_test[:,1]/2,'.',alpha=0.1)
plt.plot(y_hat_inphase,y_hat_quadrature,'.',alpha=0.1)
plt.axis('square')
plt.ylim([-2,2])
plt.xlim([-2,2])
plt.grid('on')
plt.xlabel('In-phase component')
plt.ylabel('Quadrature component')
plt.title('Representacao do diagrama de Constelacao')
# Duas polarizacoes simultaneamente

# We select a mid-power constellation
ind = 11

# Data and labels for that LOP level
DataXY = data[ind,:,:]          # We select both polarization
LabelsX = labels_X              # We select labels of X
polarization
LabelsY = labels_Y              # We select labels of Y
polarization

# Rename to the meet the variables names of sklearn
X = DataXY[:,:]                # Input values
Y = data_ideal[:,:]            # Output values

scaler = StandardScaler()
scaler.fit(X)
scaler.transform(X)
scaler.fit(Y)
scaler.transform(Y)

# Split data into train and test
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.33, random_state=42)

# Perform the regression for XY polarization

```

```

    regr_inphase_X = svm.SVR(kernel='rbf', gamma=0.5, C=1).fit(
X_train, Y_train[:,0])
    regr_quadrature_X = svm.SVR(kernel='rbf', gamma=0.5, C=1).
fit(X_train, Y_train[:,1])
    regr_inphase_Y = svm.SVR(kernel='rbf', gamma=0.5, C=1).fit(
X_train, Y_train[:,2])
    regr_quadrature_Y = svm.SVR(kernel='rbf', gamma=0.5, C=1).
fit(X_train, Y_train[:,3])

# Predict the in-phase and quadrature components
y_hat_inphase_X = regr_inphase_X.predict(X_test)
y_hat_quadrature_X = regr_quadrature_X.predict(X_test)
y_hat_inphase_Y = regr_inphase_Y.predict(X_test)
y_hat_quadrature_Y = regr_quadrature_Y.predict(X_test)

# Representation of the constellation
plt.figure()
plt.plot(y_hat_inphase_X,y_hat_quadrature_X,'o',alpha=0.1)
plt.axis('square')
plt.title('Diagrama de Constelacao para as polarizacoes X e
Y')

# We combine the in-phase and quadrature components
Y_hat_X = np.vstack((y_hat_inphase_X,y_hat_quadrature_X)).
transpose()
Y_hat_Y = np.vstack((y_hat_inphase_Y,y_hat_quadrature_Y)).
transpose()
Y_ideal_X = Y_test[:, :2]
Y_ideal_Y = Y_test[:, :2]

labels_hat = MLdetection_16QAM(Y_hat_X)
labels_ideal = MLdetection_16QAM(Y_ideal_X)

BER = sum(labels_hat!=labels_ideal)/len(labels_ideal)/4
print('A BER obtida para a polarizacao XY e:', BER)

# Representation of the constellation
plt.figure()
plt.plot(X_test[:,0]/2,X_test[:,1]/2,'o',alpha=0.1)
plt.plot(y_hat_inphase_X,y_hat_quadrature_X,'o',alpha=0.1)

```

```

plt.axis('square')
plt.ylim([-2,2])
plt.xlim([-2,2])
plt.grid('on')
plt.xlabel('In-phase component')
plt.ylabel('Quadrature component')
gamma = 1
C = 1
BER_SVR_SP_array = np.zeros(k)
BER_SVR_DP_array = np.zeros(k)
plt.figure()
for ind in range(k):
    print ('Processando',ind,'de',k-1)

    print (' Processando polarizacao X')
    DataX = data[ind,:,:2]      # We select X polarization
    X = DataX[:,:]             # Input values
    Y = data_ideal[:,0:2]      # Output values
    # Scaling
    scaler = StandardScaler()
    scaler.fit(X)
    scaler.transform(X)
    scaler.fit(Y)
    scaler.transform(Y)
    # Split data into train and test
    X_train, X_test, Y_train, Y_test = train_test_split(X,
Y, test_size=0.33, random_state=42)
    # Perform the regression
    regr_inphase = svm.SVR(kernel='rbf', gamma=gamma, C=C).
fit(X_train, Y_train[:,0])
    regr_quadrature = svm.SVR(kernel='rbf', gamma=gamma, C=
C).fit(X_train, Y_train[:,1])
    # Predict the in-phase and quadrature components
    y_hat_inphase = regr_inphase.predict(X_test)
    y_hat_quadrature = regr_quadrature.predict(X_test)
    # We combine the in-phase and quadrature components
    Y_hat = np.vstack((y_hat_inphase,y_hat_quadrature)).
transpose()
    Y_ideal = Y_test
    # Detection by maximum likelihood

```

```

labels_hat = MLdetection_16QAM(Y_hat)
labels_ideal = MLdetection_16QAM(Y_ideal)
BER_X = sum(labels_hat!=labels_ideal)/len(labels_ideal)
/4

BER_SVR_SP_array[ind] = BER_X

print(' Processando polarizacoes XY')
X = data[ind,:,:]          # Valores de entrada
Y = data_ideal[:,0:2]      # Output values
# Scaling
scaler = StandardScaler()
scaler.fit(X)
scaler.transform(X)
scaler.fit(Y)
scaler.transform(Y)
# Split data into train and test
X_train, X_test, Y_train, Y_test = train_test_split(X,
Y, test_size=0.33, random_state=42)
# Perform the regression
regr_inphase = svm.SVR(kernel='rbf', gamma=gamma, C=C).
fit(X_train, Y_train[:,0])
regr_quadrature = svm.SVR(kernel='rbf', gamma=gamma, C=
C).fit(X_train, Y_train[:,1])
# Predict the in-phase and quadrature components
y_hat_inphase = regr_inphase.predict(X_test)
y_hat_quadrature = regr_quadrature.predict(X_test)
# We combine the in-phase and quadrature components
Y_hat = np.vstack((y_hat_inphase,y_hat_quadrature)).
transpose()
Y_ideal = Y_test
# Detection by maximum likelihood
labels_hat = MLdetection_16QAM(Y_hat)
labels_ideal = MLdetection_16QAM(Y_ideal)
BER = sum(labels_hat!=labels_ideal)/len(labels_ideal)/4
BER_SVR_DP_array[ind] = BER

print('      A BER obtida para a polarizacao X e:', BER_X
)

print('      A BER obtida para as polarizacoes XY e:',
BER)

```

```

        print('      A BER ML e',BER_array[ind])

plt.figure()
P_array = np.arange(1,k+1)
plt.semilogy(P_array,BER_array)
plt.semilogy(P_array,BER_SVR_SP_array)
plt.semilogy(P_array,BER_SVR_DP_array)

plt.xlabel('Launched optical power [dBm]')
plt.ylabel('BER')
plt.grid('on')
plt.figure()
P_array = np.arange(1,k+1)
plt.semilogy(P_array,BER_array, '-.',color = 'gray' ,
markeredgecolor = 'k', label = 'ML')
plt.semilogy(P_array,BER_SVR_SP_array, '-o',color='C0',
label = "SVR - SP")
plt.semilogy(P_array,BER_SVR_DP_array, '-s',color='C1',
label = "SVR - DP")
plt.legend()

plt.xlabel('Potencia Optica de Transmissao [dBm]')
plt.ylabel('BER')
plt.grid('on')
plt.xlim([1,12])

print(P_array)
print(BER_array)
print(BER_SVR_SP_array)
print(BER_SVR_DP_array)

```