Geração de Chave na Camada Física para Sistemas FDD Baseado em Aprendizagem Profunda

Gustavo M. da Silva
Curso de Engenharia Elétrica
Universidade Federal de São Carlos
São Carlos, Brasil
gustavo.marques@estudante.ufscar.br

Diana P. Moya Osorio
Centre of Wireless Communications
University of Oulu
Oulu, Finland
diana.moyaosorio@oulu.fi

Helder Vinicius A. Galeti
Departamento de Engenharia Elétrica
Universidade Federal de São Carlos
São Carlos, Brasil
helder@ufscar.br

***Resumo*—Com o avanço das telecomunicações e a transição para a próxima geração de redes (6G), novos desafios e oportunidades emergem à superfície. A segurança na camada física (PLS, do inglês *physical layer security*) é crucial para garantir a confiabilidade da rede, especialmente considerando o crescente poder computacional de possíveis invasores. A geração de chave secreta na camada física é uma técnica de PLS, que oferece a vantagem de ser menos complexa e demandar poucos recursos, tornando-se uma grande facilitadora da segurança para sistemas com limitações computacionais, como dispositivos da internet das coisas (IoT, do inglês *internet of things*), por exemplo. No entanto, em sistemas de duplexação por divisão de frequênicia (FDD, do inglês *frequency division duplexing*), que possuem canais de *uplink* e *downlink* em diferentes frequências, a reciprocidade necessária para a geração confiável de chaves aleatórias não é diretamente aplicável devido às características distintas dos canais. Neste contexto, este trabalho propõe a utilização de técnicas de aprendizagem profunda para estabelecer uma reciprocidade artificial entre os canais de comunicação em sistemas FDD. O trabalho concentra-se na construção e treinamento de uma rede neural de quatro camadas ocultas, utilizando um extenso conjunto de dados de condições dos canais de *uplink* e *downlink* sob diferentes cenários. A partir dessa aproximação, é aplicado um esquema de geração de chave aleatória e são analisados aspectos de segurança, como a taxa de erro de chave (KER, do inglês *key error rate*) e a razão de geração de chave (KGR, do inglês *key generation ratio*). As principais contribuições do estudo são: i) a construção e treinamento da rede neural para geração de chave aleatória em sistemas FDD; ii) a análise de parâmetros da rede para garantir robustez e generalização; iii) a avaliação da segurança das chaves geradas utilizando métricas como KER e KGR.**

***Index Terms*—aprendizagem profunda, duplexação por divisão de frequência, geração de chave aleatória na camada física, inteligência artificial, segurança na camada física.**

I. Introdução

Ao longo dos anos, a humanidade busca por formas de se conectar uns aos outros, trazendo assim ao mundo o advento das telecomunicações, que ao longo do tempo vem sido amplamente estudada e pode ser definida por algumas gerações distintas cujos problemas resolvidos e paradigmas trazidos foram elaborados e investigados na comunidade científica em larga escala [1]. Nos dias atuais, à medida que se realiza a implementação da quinta geração de telecomunicações (5G), emerge na academia as discussões a respeito da sexta geração (6G), juntamente com seus desafios e oportunidades, trazendo agora não mais apenas um aperfeiçoamento na internet das coisas proposta pelo 5G ou na conectividade à internet global como em gerações anteriores, mas sim uma ruptura de paradigmas ao utilizar de conceitos de inteligência artificial, teoria de jogos, aprendizagem de máquina, entre outros, para dar cabo a problemas e pontas soltas há muito conhecidos [2].

Uma vez que a próxima geração de telecomunicações emerge com um altíssimo padrão de segurança da informação, que a cada dia vem ganhando mais importância inclusive na sociedade brasileira, é de extrema criticidade garantir a segurança também em camadas mais baixas e simples do modelo de interconexão de sistemas abertos (OSI, do inglês *open systems interconnection*), dado que uma quantidade massiva de informações sensíveis serão transmitidas e recebidas através dessas redes. Nesse contexto, a segurança na camada física (PLS, do inglês *physical layer security*) desempenha um papel fundamental. À medida que as habilidades computacionais dos possíveis espiões na rede são cada vez mais elevadas, abordar a segurança na camada física torna-se uma maneira de garantir a confiabilidade da rede, sem depender exclusivamente de complexos algoritmos de criptografia e altos recursos computacionais [3].

Por sua vez, o estudo da segurança na camada física é bastante extenso, e conta com uma gama infindável de aplicações a fim de garantir a confidencialidade, autenticidade e disponibilidade das informações. Entre essas aplicações, destaca-se a importância da geração de chave aleatória para reforçar a segurança nessa camada [4]. A geração de chaves aleatórias desempenha um papel crucial na proteção das comunicações, pois complementa técnicas convencionais e avançadas de criptografia, podendo ser uma forma de prevenir ataques por impersonalização, uma vez que utiliza características intrínsecas do canal físico. A aleatoriedade das chaves é essencial para garantir que não haja padrões previsíveis que possam ser explorados por potenciais invasores. Além disso, a geração de chave aleatória na camada física oferece uma camada adicional de segurança, uma vez que é mais difícil para um atacante interceptar ou manipular a chave durante a transmissão [5]. Dessa forma, a geração de chave aleatória na camada física contribui significativamente para a proteção dos sistemas de comunicação, fortalecendo a confidencialidade e a integridade dos dados transmitidos.

Para a aplicação das técnicas de geração de chaves na camada física, emergem três princípios de fundamental importância, os quais devem ser rigorosamente observados e essencialmente incorporados ao sistema em consideração. O primeiro princípio destaca-se pela relevância da aleatoriedade sob variação temporal, implicando na necessidade de oscilações nos canais de comunicação ao longo do tempo, alinhados com os princípios que regem as propriedades das ondas eletromagnéticas. O segundo princípio concentra-se na variação espacial, pois a existência do canal está intrinsecamente relacionada à distância específica entre os nós interconectados na rede. O terceiro princípio, a reciprocidade dos canais, assume papel crucial ao garantir que a codificação e decodificação da informação, dentro de um sistema predeterminado, sejam satisfatoriamente viabilizadas, sendo assim indispensável a presença de características recíprocas nos canais subjacentes, nos quais as chaves aleatórias podem ser geradas com êxito [6]. Este conjunto de princípios compõe o arcabouço conceitual que respalda a segurança e eficácia da geração de chaves no contexto da camada física [7].

Por outro lado, a geração de chaves aleatórias para sistemas de duplexação por divisão de frequência (FDD, do inglês *frequency division duplexing*) apresenta desafios adicionais devido à natureza dos canais de *uplink* e *downlink* estarem em diferentes frequências, resultando na ausência de características recíprocas. A reciprocidade é um atributo crucial para a geração de chaves aleatórias confiáveis, pois permite que as informações obtidas em uma direção (como o canal de *uplink*) sejam usadas para gerar chaves que também sejam aplicáveis na direção oposta (como o canal de *downlink*) [4]. No entanto, nos sistemas FDD, as características desses canais podem ser bastante distintas devido às diferenças nas frequências de operação e nas condições de propagação. Isso dificulta a aplicação direta de técnicas de geração de chaves aleatórias baseadas em reciprocidade, exigindo abordagens alternativas para garantir a segurança na camada física desses sistemas [8]. Nesse contexto, torna-se essencial desenvolver métodos eficientes e adaptados para a geração de chaves aleatórias em sistemas FDD, levando em consideração as particularidades dos canais de comunicação envolvidos e mantendo um alto nível de segurança para proteger as comunicações nessas configurações específicas.

Paralelamente, diversas soluções de inteligência artificial vêm ganhando enfoque na resolução de problemas complexos em várias áreas, e as telecomunicações não são exceção [9]. A aprendizagem profunda (em inglês, *deep learning*) é uma subárea da aprendizagem de máquina (em inglês, *machine learning*) que revolucionou a forma de se lidar com dados e problemas complexos. A aprendizagem de máquina refere-se a um conjunto de técnicas que permitem que os computadores aprendam padrões e tomem decisões com base em dados, sem serem explicitamente programados. Ela envolve algoritmos que identificam relações e estruturas nos dados, permitindo que os sistemas ”aprendam”e melhorem seu desempenho ao longo do tempo [10]. Nesse sentido, a aprendizagem profunda, se concentra em redes neurais artificiais compostas por várias camadas. Essas redes são capazes de aprender representações complexas e abstratas dos dados, permitindo o processamento de informações de forma hierárquica [11].

No contexto específico dos sistemas FDD, técnicas avançadas de aprendizagem profunda têm se mostrado promissoras para abordar a questão de reciprocidade. A aplicação de algoritmos de *deep learning* permite explorar os padrões e características intrínsecas de um dos canais de comunicação para que se possa prever o outro canal, a fim de construir uma reciprocidade artificial entre esses canais, podendo então aplicar os esquemas de geração de chave aleatória na camada física que são altamente dependentes dessa reciprocidade. Este trabalho, concentra-se assim na construção de uma rede neural de 4 camadas ocultas que possa ser treinada para, através de um conjunto extenso de dados das condições dos canais de *uplink* e *downlink* de um sistema FDD sob diferentes cenários, ser capaz de estabelecer características recíprocas entre estes canais. A partir disto, será aplicado um esquema de geração de chave aleatória e serão quantificados e discutidos os aspectos de segurança desta chave, através de métricas como a taxa de erro de chave (KER, do inglês *key error rate*) e a razão de geração de chave (KGR, do inglês *key generation ratio*). As principais contribuições deste trabalho são: i) a construção e treinamento de uma rede neural aplicada à geração de chave aleatória para sistemas FDD; ii) a análise de parâmetros da rede para garantir a robustez e generalização das aproximações; iii) a análise das chaves aleatória gerada a partir da aproximação sob métricas de KER e KGR.

II. Trabalhos Relacionados

Nos últimos anos, várias técnicas de geração de chaves aleatórias têm sido desenvolvidas para sistemas FDD. Uma dessas técnicas envolve a extração de parâmetros recíprocos do canal que sejam independentes da frequência para gerar a chave. No entanto, a obtenção precisa desses parâmetros requer recursos significativos, como largura de banda e múltiplas antenas [12]. Nesse mesmo sentido, também foi proposto um método baseado na reciprocidade dos autovalores da matriz de covariância do canal em [13], porém esta solução exige uma configuração especial do arranjo de antenas. Outra abordagem envolve o estabelecimento de um canal de *downlink* com ganho de canal recíproco ao canal de *uplink*, realizando uma medição de canal especial para determinação do ganho, sendo conhecidos como métodos baseados em *loopback* [14]. Esses métodos requerem detecção de canal complexa e podem aumentar o risco de espionagem [15]. Um terceiro método, baseia-se na construção de características recíprocas a partir do conhecimento prévio do modelo de canal realizando uma separação das frequências nos possíveis caminhos. Contudo, esta solução é especialmente desafiadora em ambientes complexos e de múltiplos caminhos (em inglês, *multipath*) [16].

Diante das limitações das abordagens propostas acima, Zhang em [17] propõe um novo método de geração de chaves para sistemas FDD usando aprendizado profundo. O método proposto utiliza a rede neural para aprender a função de mapeamento entre diferentes bandas de frequência para

construir características recíprocas de canal. Diferentemente das abordagens anteriores, não é necessária a implementação de *loopback*, tornando o método mais flexível e evitando a necessidade de cálculos complexos para extrair parâmetros recíprocos dos canais. Este trabalho baseia-se nos estudos de Zhang e propõe a percepção da rede em alterações de parâmetros específicos essenciais para a comunicação massiva.

III. Modelo do Sistema

O modelo sob estudo nesse trabalho foca na comunicação entre dois usuários legítimos da rede, nomeados por Alice (A), que representa uma estação rádio-base de transmissão e Bob (B), que representa um usuário da rede que deseja se comunicar com Alice com segurança na presença de um espião passivo Eve (E), localizado a uma distância de d metros de Bob.

A. Modelo do Canal

Uma vez que Alice e Bob irão comunicar-se seguindo um esquema de geração de chaves aleatórias na camada física, então é de fundamental importância que sejam enviados durante o processo de comunicação sinais piloto alternadamente para que seja realizada a estimativa do canal de Bob por Alice e vice-versa, obtendo assim as estimativas $\hat{h}_A$ e $\hat{h}_B$. Para sistemas de duplexação por divisão de tempo (TDD, do inglês *time division duplexing*), garantir a reciprocidade do canal é uma tarefa relativamente simples, uma vez que durante o tempo de coerência as frequências dos canais de *downlink* e *uplink* permanecem iguais. Sendo assim, é possível projetar um protocolo de geração de chave $\mathcal{K}(\cdot)$, que converterá o sinal analógico em binário gerando as chaves

$$K_A = \mathcal{K}(\hat{h}_A) \tag{1}$$

$$K_B = \mathcal{K}(\hat{h}_B) \tag{2}$$

Contudo, a aplicação direta desses protocolos projetados para sistemas TDD não é eficaz quando aplicado a sistemas FDD, uma vez que as condições de reciprocidade não estão mais presentes, dado que a frequência de *downlink* e *uplink* são diferentes independentemente do instante de tempo.

Sendo assim, a aplicação de aprendizagem profunda pode, primeiro garantir uma reciprocidade gerada artificialmente para os sistemas FDD, para então possibilitar a aplicação dos conceitos de geração de chave aleatória na camada física que são aplicados para os sistemas TDD.

Dado o cenário apresentado acima, para este trabalho será considerado que Alice e Bob estão equipados com apenas uma antena, trabalhando no modo FDD, ou seja, transmitindo informações simultaneamente em diferentes frequências f_{BA} e f_{AB}, que representam a comunicação Bob $\rightarrow$ Alice (Banda 1) e Alice $\rightarrow$ Bob (Banda 2), respectivamente.

A resposta do canal ao impulso (CIR, do inglês *channel impulse response*) é uma métrica muito utilizada para modelar os canais de comunicação no tempo e é essencial para projetar as chaves aleatórias de um sistema, o CIR representa a resposta combinada dos canais de *uplink* e *downlink* a uma entrada de impulso unitário e pode ser obtido, considerando N possíveis caminhos no ambiente, por

$$h(f,\tau) = \sum_{n=0}^{N-1} \alpha_n e^{-j2\pi f \tau_n + j\phi_n} \delta(\tau - \tau_n) \tag{3}$$

em que f denota a frequência de portadora, e α_n, τ_n e ϕ_n denotam respectivamente a magnitude, atraso de tempo e atraso de fase no n-ésimo caminho. É importante observar que α_n depende da distância entre Alice e Bob, da frequência de portadora f e das características do espalhamento de ondas do ambiente. ϕ_n por sua vez depende dos espalhadores de onda presentes no ambiente e também dos ângulos de incidência nestes materiais em um determinado caminho. Por fim, $\tau_n = d_n/c$ depende apenas da distância entre Alice e Bob e da velocidade da luz.

Já a representação do canal no domínio da frequência é obtida pela resposta do canal à frequência (CFR, do inglês *channel frequency response*) e mostra como o canal afeta diferentes componentes de frequência do sinal. O CFR é obtido através da Transformada de Fourier do CIR, e pode ser representado, para múltiplas frequências subportadoras por

$$H(f,l) = \sum_{n=0}^{N-1} \alpha_n e^{-j2\pi f \tau_n + j\phi_n} e^{-j2\pi f_l \tau_n} \tag{4}$$

em que f_l é a frequência da l-ésima subportadora relativa à frequência central f. Define-se então o vetor de canal $\mathbf{H}(f)$ de dimensão $1 \times L$, cujos elementos serão definidos como $\mathbf{H}(f) = \{H(f,0), H(f,1), ..., H(f,L-1)\}$ representando o CFR na frequência f, em que L é o número total de subportadoras. Portanto pode-se definir $\mathbf{H_1} = \mathbf{H}(f_{BA})$ como o CFR da Banda 1 e $\mathbf{H_2} = \mathbf{H}(f_{AB})$ como o CFR da Banda 2.

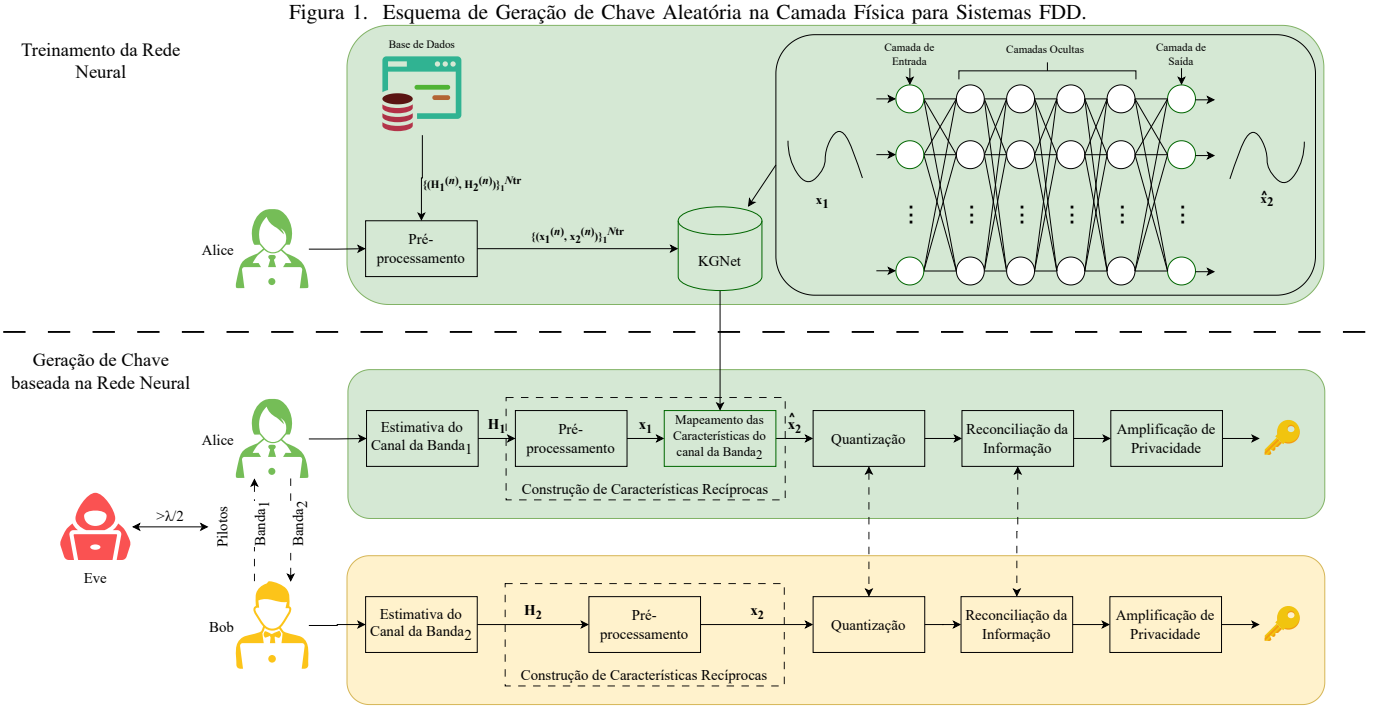
B. Modelo de Ataque

Com base nas suposições da maioria dos esquemas de geração de chaves [17]–[19], este trabalho se concentra no espião passivo, muitas vezes intrínseco à natureza *broadcast* da rede de comunicação. O espião, chamado Eve, está localizado a uma distância d dos usuários legítimos, maior que metade do comprimento de onda das bandas 1 e 2, o que pode ser expresso matematicamente como:

$$d > \max\left\{\frac{c}{2f_{AB}}, \frac{c}{2f_{BA}}\right\} \tag{5}$$

Dado que os ganhos do canal sem fio perdem totalmente sua correlação a uma distância superior à metade do comprimento de onda em ambientes de multipercurso, assume-se que o canal de Eve é independente do canal dos usuários legítimos. Portanto, Eve não pode inferir o canal entre os usuários legítimos para gerar a chave aleatória a partir do canal que chega a ele.

Figura 1. Esquema de Geração de Chave Aleatória na Camada Física para Sistemas FDD.



Fonte: Autor.

IV. Construção de Características Recíprocas

Como mencionado nas seções anteriores, a implementação dos protocolos de geração de chaves aleatórias na camada física desenvolvida para os sistemas TDD não pode ser diretamente extrapolada para os sistemas FDD, devido à ausência de reciprocidade no canal. Assim, o objetivo deste trabalho é desenvolver uma rede neural na parte de Alice que, através de um processo de treinamento *offline*, seja capaz de prever as estimativas de canal obtidas por Bob. O propósito é criar artificialmente características recíprocas entre as estimativas de canal desses dois usuários legítimos, permitindo, desse modo, a geração confiável e decodificável de uma chave aleatória para a comunicação de ponta a ponta.

Para alcançar esse objetivo, foi proposta a implementação do esquema de geração de chave aleatória ilustrado na Figura 1. Esse esquema descreve um processo em cinco etapas distintas.

A. Estimativa de Canal

O ponto de partida do esquema para a criação da chave aleatória consiste na etapa inicial de estimativa do canal. Essa etapa é baseada na comunicação entre Alice e Bob, onde eles alternadamente enviam sinais piloto para permitir a extração conjunta da informação sobre o estado do canal (CSI, do inglês *channel state information*). Através dessa estimativa de canal, tanto Alice quanto Bob conseguem obter os vetores de coeficientes do canal para as bandas de frequência 1 e 2, que são denotados como $\mathbf{H_1}$ e $\mathbf{H_2}$, conforme definido na Equação 4.

B. Construção de Características Recíprocas

Após a conclusão da estimativa dos estados dos canais, tanto Alice quanto Bob derivam características recíprocas dos CSIs, marcando essa etapa como de máxima importância para o esquema de geração de chaves.

No caso de Bob, essa etapa envolve exclusivamente o pré-processamento do vetor de coeficientes de canal $\mathbf{H_2}$. Esse pré-processamento abrange a separação das partes real e imaginária dos coeficientes de canal, seguida pela normalização desses valores para o intervalo entre 0 e 1. Portanto, o pré-processamento pode ser compreendido como uma função que recebe um vetor $\mathbf{H_2}$ de dimensão $1 \times L$, composto por números complexos, e produz um vetor $\mathbf{x_2}$ de dimensão $1 \times 2L$, contendo elementos reais no intervalo $[0, 1]$.

Por outro lado, para Alice esta etapa é formada por dois momentos distintos:

1) Pré-processamento: O primeiro momento é o pré-processamento do vetor de coeficientes de canal $\mathbf{H_1}$, que segue o mesmo algoritmo de pré-processamento utilizado por Bob, supracitado. Isso resulta no vetor $\mathbf{x_1}$.
2) Mapeamento das Características da Banda 2: O segundo momento consiste no mapeamento das características da banda 2, empregando uma rede neural previamente treinada. Nesse sentido, Alice utiliza a rede neural treinada para, a partir do resultado do seu próprio pré-processamento $\mathbf{x_1}$, obter uma estimativa do canal de Bob após o pré-processamento. A estimativa do vetor $\mathbf{x_2}$ que Alice obteve com apoio da rede neural será denotada por $\hat{\mathbf{x}}_\mathbf{2}$.

Por meio dessas etapas, Alice e Bob podem obter, respec-

tivamente, o vetor de características do canal $\hat{\mathbf{x}}_\mathbf{2}$ e $\mathbf{x_2}$ de dimensão $1 \times 2L$.

C. Quantização

Após obter as características do canal, Alice e Bob devem aplicar o mesmo algoritmo de quantização para converter essas características em um fluxo de bits binários com uma baixa taxa de erro de chave (KER, do inglês *key error rate*). Neste trabalho é proposto um método de quantização baseado em distribuição Gaussiana com banda de guarda (GDQG, do inglês *Gaussian distribution-based quantization with guard-band*), bastante utilizado para sistemas FDD.

Para este método de quantização é fundamental que sejam determinadas a média e a variância das partes real e imaginária do CFR considerando as frequências subportadoras. Esses valores podem ser expressos como

$$\mu = \frac{1}{2L} \sum_{l=0}^{2L-1} x^l \tag{6}$$

$$\sigma^2 = \frac{1}{2L-1} \sum_{l=0}^{2L-1} \left(x^l - \mu\right)^2 \tag{7}$$

em que x^l denota o l-ésimo elemento do vetor $\mathbf{x}$.

A distribuição das características do canal pode ser aproximada por uma distribuição Gaussiana. Portanto, ajustou-se a probabilidade das características do canal a uma distribuição Gaussiana definida como $\mathcal{N}_\mathcal{Q} = \mathcal{N}(\mu, \sigma^2)$.

O k-ésimo intervalo de quantização é calculado como

$$\left[F^{-1}\left(\frac{k-1}{K} + \varepsilon\right), F^{-1}\left(\frac{k}{K} - \varepsilon\right)\right], k = 2, ..., K-1 \tag{8}$$

cujo primeiro intervalo de quantização é $[0, F^{-1}(\frac{1}{K} - \varepsilon)]$ e o último é $[F^{-1}(\frac{K-1}{K} + \varepsilon), 1]$, onde $F^{-1}(\cdot)$ é definido como o inverso da função de distribuição acumulada (CDF, do inglês *cumulative distribution function*) de $\mathcal{N}_\mathcal{Q}$. Ademais, K representa o nível de quantização e $\varepsilon \in (0, 1/2K)$ é definido como o fator de quantização, usado para estabelecer o limite da banda de guarda. Em seguida, foi usada a codificação binária comum para converter os valores em um fluxo de bits, onde todas as características que não estão nos intervalos de quantização são definidas como -1.

Os passos do método de quantização são fornecidos no Algoritmo 1. Por fim, Alice e Bob enviam um ao outro os índices cujos valores são -1 e excluem todos esses bits, obtendo assim as chaves iniciais Q_A e Q_B.

D. Reconciliação da Informação

Na etapa de reconciliação da informação, os nós da rede compararão as chaves brutas geradas independentemente, identificando e corrigindo discrepâncias entre elas. Essas discrepâncias podem ser resultado de ruídos, erros de medição ou outras imperfeições no canal de comunicação. O objetivo é garantir que as chaves geradas por ambos os dispositivos sejam coerentes e compatíveis antes de prosseguir para a próxima etapa.

Algoritmo 1 Processo de Quantização

Entrada: O vetor de características do canal x, o fator de quantização ε

Saída: A sequência binária quantizada Q

```
Calcular a média, μ e a variância, σ² do vetor x;
Construir a distribuição Gaussiana N_Q = N(μ, σ²);
Calcular a inversa da CDF de N_Q;
Declarar ε;
Calcular os intervalos de quantização conforme a Equação 8;
para i = 0 : 2L − 1 faça
    se x_i está em um dos intervalo de quantização então
        Codificação Binária;
    senão
        Codificar como −1;
    fim se
fim para
retorne Q;
```

E. Amplificação de Privacidade

Após a reconciliação das informações, as chaves podem conter informações redundantes ou informações que ainda não são suficientemente secretas. A etapa de amplificação de privacidade é projetada para resolver esse problema, eliminando qualquer informação redundante e reduzindo o tamanho efetivo da chave, ao mesmo tempo em que aumenta sua aleatoriedade.

Neste trabalho não serão trabalhados algoritmos de reconciliação de informação ou amplificação de privacidade, concentrando-se apenas na eficácia da geração da chave inicial a partir da implementação da rede neural.

V. Métricas de Análise

Neste trabalho, será adotado o uso do erro médio quadrático normalizado (NMSE, do inglês "normalized mean squared error") como métrica para avaliar a acurácia da predição da rede neural. O NMSE é definido por

$$NMSE = E\left[\frac{||\hat{\mathbf{x_2}} - \mathbf{x_2}||_2^2}{||\mathbf{x_2}||_2^2}\right] \tag{9}$$

em que $E[\cdot]$ representa o operador esperança e $||\cdot||_2$ denota a norma euclidiana. A partir dessa métrica, o desempenho da chave inicial é avaliado utilizando duas outras métricas: a taxa de erro da chave (KER, do inglês "key error rate") e a taxa de geração da chave (KGR, do inglês "key generation rate"). A KER é definida como a proporção entre o número de bits divergentes entre duas chaves e o número total de bits. Já a KGR é definida como a proporção entre o número de bits válidos de uma chave inicial e o número de subportadoras. Caso cada característica real e imaginária das subportadoras contribua para a geração de um bit válido da chave, e a banda de guarda não seja utilizada durante a quantização, a KGR alcança um valor máximo de 2.

VI. Construção da Rede Neural

A construção da rede neural é fundamentada em princípios essenciais da aprendizagem profunda, resultando na criação de uma rede com quatro camadas ocultas. Para uma exposição mais detalhada das etapas envolvidas na elaboração da rede neural, esta seção será dividida em segmentos que abordam a aquisição dos conjuntos de treinamento e teste, o processo de pré-processamento dos dados, uma revisão da arquitetura da rede neural, além de uma explicação dos estágios de treinamento e teste. O código desenvolvido em linguagem *Python* para a construção da rede neural utilizada pode ser acessado em [20].

A. Aquisição dos conjuntos de treinamento e teste

Devido à complexidade e variabilidade dos ambientes, existe uma infinidade de diferentes ambientes possíveis. No entanto, é impraticável criar conjuntos de dados para cada um desses ambientes. Neste trabalho, optou-se por considerar um ambiente específico.

Em um ambiente específico, o CSI das bandas de frequência 1 e 2 é tratado como o conjunto original de dados, $\mathbb{D}_O$. Esse conjunto é então dividido em conjuntos de treinamento e de teste, representados por $\mathbb{D}_{OTr}$ e $\mathbb{D}_{OTe}$, respectivamente, e definidos como

$$\mathbb{D}_{OTr} = \left\{ \left(\mathbf{H_1}^{(n)}, \mathbf{H_2}^{(n)} \right) \right\}_{n=1}^{N_{Tr}} \tag{10}$$

$$\mathbb{D}_{OTe} = \left\{ \left(\mathbf{H_1}^{(n)}, \mathbf{H_2}^{(n)} \right) \right\}_{n=1}^{N_{Te}} \tag{11}$$

em que N_{Tr} e N_{Te} representam a quantidade de amostras nos conjuntos de treinamento e teste, respectivamente. É importante destacar que o conjunto de treinamento não precisa incluir todos os possíveis canais entre Alice e Bob, mas deve ser suficientemente representativo para permitir que a rede neural treinada generalize de maneira precisa e acurada sua resposta.

B. Pré-processamento dos dados

Conforme explicado na subseção IV-B, esta etapa é crucial no processo de geração de chaves aleatórias e tem um impacto significativo no aprimoramento do desempenho da rede neural. Nesse sentido, adotou-se duas funções distintas:

1) A primeira função é encarregada de separar os valores complexos dos coeficientes do canal em suas partes real e imaginária. Essa função é definida como:

$$\xi^{(1)} : \mathbf{H}' \rightarrow (\mathcal{R}(\mathbf{H}), \mathcal{I}(\mathbf{H})) \tag{12}$$

em que $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ denotam a parte real e imaginária de um matriz, vetor ou escalar, respecticamente. Após a aplicação dessa função, os vetores complexos de coeficientes do canal, com dimensão $1 \times L$, pertencentes às bandas de frequência 1 e 2, são transformados em vetores de valores reais com dimensão $1 \times 2L$, denotados por $\mathbf{H}'_\mathbf{1}$ e $\mathbf{H}'_\mathbf{2}$, respectivamente.]

2) A segunda função desempenha a normalização, cuja responsabilidade é dimensionar os valores dos elementos dos vetores para o intervalo entre 0 e 1. Para realizar isso, os valores mínimo e máximo do vetor são estabelecidos por

$$\begin{cases} H'^{l}_{1,\max} = \max\limits_{n=1,\ldots,N_{Tr}} \{H'^{l}_1\}^n \\ H'^{l}_{1,\min} = \min\limits_{n=1,\ldots,N_{Tr}} \{H'^{l}_1\}^n \end{cases} \quad l = 0, ..., 2L-1 \tag{13}$$

em que $\{H'^{l}_1\}$ é o l-ésimo elemento de $\mathbf{H}'_\mathbf{1}$. Portanto, a função de normalização é descrita por

$$\xi^{(2)} : x^l_1 \rightarrow \frac{H'^{l}_1 - H'^{l}_{1,\min}}{H'^{l}_{1,\max} - H'^{l}_{1,\min}}, \; l = 0, ..., 2L-1 \tag{14}$$

em que x^l_1 é o l-ésimo elemento de $\mathbf{x_1}$. O procedimento de normalização é igualmente aplicado a $\mathbf{H}'_\mathbf{2}$. Após essa etapa de processamento, os conjuntos de treinamento e teste podem ser definidos como $\mathbb{D}_{T_r} = \left\{ \left(\mathbf{x_1}^{(n)}, \mathbf{x_2}^{(n)} \right) \right\}_{n=1}^{N_{Tr}}$ e $\mathbb{D}_{Te} = \left\{ \left(\mathbf{x_1}^{(n)}, \mathbf{x_2}^{(n)} \right) \right\}_{n=1}^{N_{Te}}$, respectivamente. Esses conjuntos podem ser diretamente fornecidos como entradas para a rede neural. Deve-se ressaltar que os valores $H'^{l}_{1,\max}$, $H'^{l}_{1,\min}$, $H'^{l}_{2,\max}$ e $H'^{l}_{2,\min}$ utilizados para a normalização dos conjuntos de treinamento e teste são obtidos exclusivamente a partir do conjunto de treinamento.

C. Arquitetura da Rede Neural

Baseado em uma rede neural de alimentação direta (em inglês *feedforward network*), é proposta a criação de um KGNet, isto é, uma rede neural específica para a geração de chaves aleatórias na camada física, composta por uma camada de entrada, quatro camadas ocultas e uma camada de saída. Conforme ilustrado na Figura 1, a entrada da rede é $\mathbf{x_1}$, obtida após o pré-processamento de $\mathbf{H_1}$. A saída da rede é uma cascata de transformações não lineares de $\mathbf{x_1}$ construídas inteiramente pela rede neural, ou seja,

$$\hat{\mathbf{x_2}} = f(\mathbf{x_1}, \Omega) \tag{15}$$

em que Ω representa todos os parâmetros ajustáveis nessa rede, que é utilizada para resolver um problema de regressão vetorial. Nas camadas ocultas e na camada de saída, empregam-se funções de ativação específicas. A função de ativação utilizada nas camadas ocultas é a função unidade linear retificada (ReLU, do inglês *rectified linear unity function*), enquanto na camada de saída, adota-se a função sigmoide. Ambas funções são amplamente empregadas em redes neurais para abordar problemas de regressão.

A configuração das camadas de neurônios é definida considerando o número de subportadoras como critério. Para otimizar o processamento, o número de neurônios nas camadas de entrada e saída é estabelecido como o dobro do número de subportadoras. Isso se deve à necessidade de separar as partes real e imaginária de cada valor de subportadora durante a etapa de pré-processamento. No contexto específico deste projeto, que envolve um sistema com 64 frequências subportadoras,

as camadas ocultas são dimensionadas com 512 neurônios na primeira e última camada, e 1024 neurônios nas duas camadas intermediárias.

Consequentemente, a configuração das camadas para um canal com 64 subportadoras resulta em um total de 128 neurônios na camada de entrada, seguidos por 512, 1024, 1024, 512 e novamente 128 neurônios na camada de saída. Essa estrutura visa aprimorar o desempenho geral do canal por meio de uma distribuição adequada dos neurônios nas diferentes etapas de processamento da rede neural.

D. Treinamento e teste

Durante a fase de treinamento, o conjunto $\mathbb{D}_{Tr}$ é considerado como o conjunto completo de treinamento. A cada intervalo de tempo conhecido como "época", um conjunto de amostras é selecionado aleatoriamente a partir do conjunto de treinamento. A rede neural é então treinada com o objetivo de minimizar a discrepância entre a saída $\hat{\mathbf{x_2}}$ e a entrada $\mathbf{x_2}$. Para essa finalidade, é empregado pela rede neural um algoritmo de estimação adaptativa de momentos (ADAM, do inglês *adaptive moment estimation*). ADAM é um algoritmo de otimização usado para ajustar os parâmetros de uma rede neural durante o treinamento, combinando conceitos de otimização de gradiente estocástico (SGD, do inglês *stochastic gradient descent*) com estimação de momentos adaptativos [21]. O objetivo principal do algoritmo ADAM é efetivamente otimizar os parâmetros de uma rede neural, buscando a convergência mais rápida possível com uma taxa de aprendizado adaptativa para cada parâmetro. É relevante destacar que a função NMSE é empregada como função de perda da rede neural, conforme abordado na seção V.

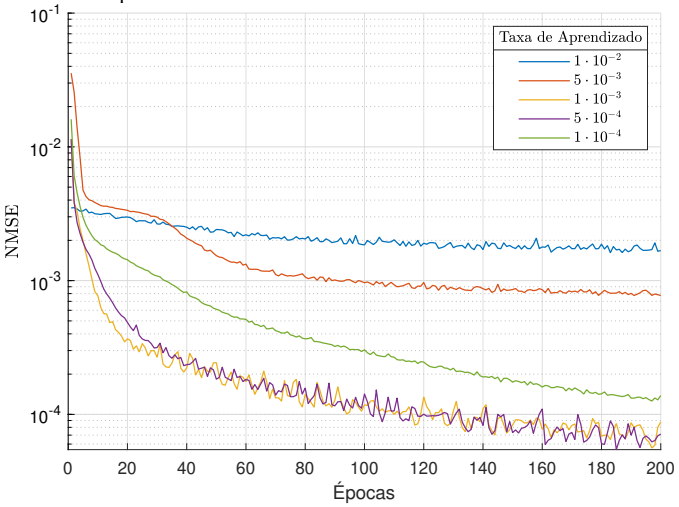
No estágio de testes, os parâmetros da rede neural permanecem fixos e o conjunto de testes $\mathbb{D}_{Te}$ é utilizado para testar a performance da rede após o processo de treinamento. Depois de concluídas todas as iterações, o conjunto de testes pode ser submetido à quantização, sendo empregado na avaliação da performance da rede neural conforme as métricas de KER e KGR.

VII. Resultados Numéricos e Discussões

Inicialmente, é essencial abordar novamente a questão da obtenção dos conjuntos de treinamento e teste, conforme discutido na subseção VI-A. No intuito de adquirir o conjunto $\mathbb{D}_O$, foi adotado o ambiente oferecido pelo cenário "I1"fornecido pelo DeepMIMO. Esse cenário consiste em uma configuração de múltiplas entradas e múltiplas saídas (MIMO, do inglês *multiple-input-multiple-output*) com um grande número de usuários em um ambiente *indoor* [22]. O modelo é composto por uma estação base (EB) que pode ser configurada de modo a ter uma única antena ativa e até 100000 usuários.

Nesse contexto, a EB foi associada à Alice, enquanto os diversos usuários potenciais representaram possíveis posições para Bob e Eve. O cenário oferece disponibilidade em duas frequências operacionais, $2,4$ GHz e $2,5$ GHz, correspondentes às frequências de portadora das bandas 1 e 2. O número de subportadoras foi fixado em 64, o número de caminhos para a propagação das ondas em 5, e a largura de banda em $0,5$ GHz, a fim de gerar o conjunto de dados. Esse conjunto de dados abrange os canais entre cada possível posição para Bob e Eve, bem como a única antena para as bandas 1 e 2. Para constituir os conjuntos de treinamento e teste, até 100000 podem ser selecionados aleatoriamente e posteriormente divididos entre esses conjuntos.

Figura 2. Variação no NMSE ao longo de 200 épocas, para diferentes valores de taxas de aprendizado.

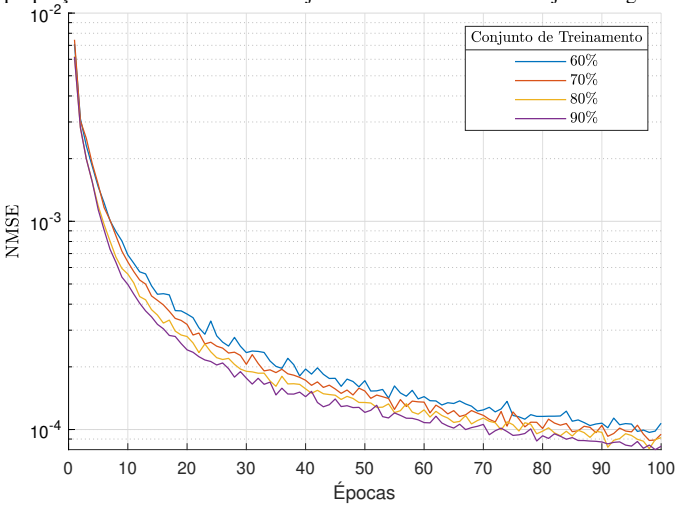


Considerando um total de 20 mil usuários, dos quais 80% são destinados ao treinamento da rede neural. Nesta etapa ainda não se faz necessário o uso de quantização. Fonte: Autor.

Dessa forma, a Figura 2 ilustra a variação da função de perda ao longo das épocas, considerando diferentes valores de taxa de aprendizado. A taxa de aprendizado em uma rede neural desempenha um papel fundamental, pois regula o tamanho dos passos que os algoritmos de otimização dão em direção ao gradiente da função de perda durante o ajuste dos pesos da rede. Em essência, a taxa de aprendizado influencia o quanto os pesos da rede são atualizados em relação ao gradiente calculado. Uma taxa de aprendizado excessivamente baixa pode resultar em convergência lenta, enquanto uma taxa muito alta pode fazer com que a rede fique presa em um ciclo iterativo (*loop*), impedindo a convergência para o mínimo da função.

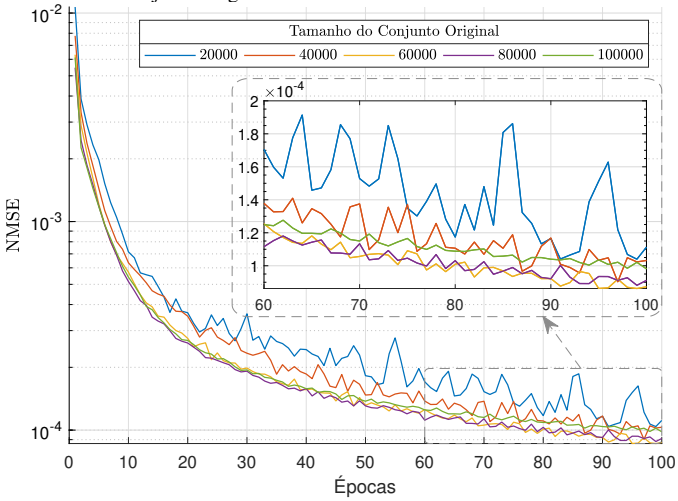
Observa-se que as curvas associadas às taxas de aprendizado de $0,01$ e $0,005$ convergem para valores consideravelmente elevados, sugerindo que essas taxas podem levar a um *loop* próximo a um mínimo local, tornando difícil a saída desse mínimo. A curva correspondente à menor taxa de aprendizado, $0,0001$, apresenta uma queda extremamente lenta, indicando que este valor é excessivamente baixo para o aprendizado. Por fim, as curvas referentes às taxas de aprendizado de $0,001$ e $0,005$ demonstram um desempenho semelhante, indicando que essas taxas são próximas da taxa de aprendizado ideal para a rede neural, podendo ser consideradas padrões para futuras simulações. Isso ressalta a importância de selecionar cuidadosamente a taxa de aprendizado apropriada, pois ela

Figura 3. Variação no NMSE ao longo de 100 épocas, para diferentes proporções entre o tamanho do conjunto de treinamento e o conjunto original.



Considerando um total de 60 mil usuários. Nesta etapa ainda não se faz necessário o uso de quantização. Fonte: Autor.

Figura 4. Variação no NMSE ao longo de 100 épocas, para diferentes tamanhos do conjunto original.



Considerando que 80% desse conjunto é destinado ao treinamento da rede neural. Nesta etapa ainda não se faz necessário o uso de quantização. Fonte: Autor.

influencia significativamente a eficiência do processo de treinamento da rede neural. Ademais, através desta figura pode-se observar que não há ganhos significativos ao longo de mais de 100 épocas. Diante disso, a decisão foi tomada de diminuir a quantidade de épocas nas simulações subsequentes. Essa abordagem visa não apenas reduzir o tempo de simulação, mas também permite direcionar esse tempo para um aumento no tamanho do conjunto de dados original.

A Figura 3, por sua vez denota a queda da função de perda ao longo das épocas para diferentes proporções entre o tamanho do conjunto de treinamento, $\mathbb{D}_{OTr}$ e o tamanho do conjunto original, $\mathbb{D}_O$. É intuitivo que quanto mais dados forem utilizados para treinar a rede neural, maior será a confiabilidade na generalização dos resultados. No entanto, é possível observar que quando a taxa de aprendizagem está bem ajustada e o tamanho do conjunto original é suficiente, o aumento do tamanho do conjunto de treinamento não causou um impacto significativo na redução da função de perda para a rede proposta. Em média, a diferença entre a melhor e a pior curva é de aproximadamente $2,55 \cdot 10^{-5}$. Isso indica que é possível optar por um conjunto de treinamento menor, o que resulta em uma economia de tempo de simulação. Decidiu-se definir a proporção em 80% para as simulações subsequentes, imaginando um cenário em que a rede neural seja treinada apenas uma vez garantindo uma qualidade maior na generalização, ainda que pequena. Sendo assim, a partir deste ponto, os seguintes resultados irão considerar a implementação da rede neural, levando em conta os dados do conjunto de testes.

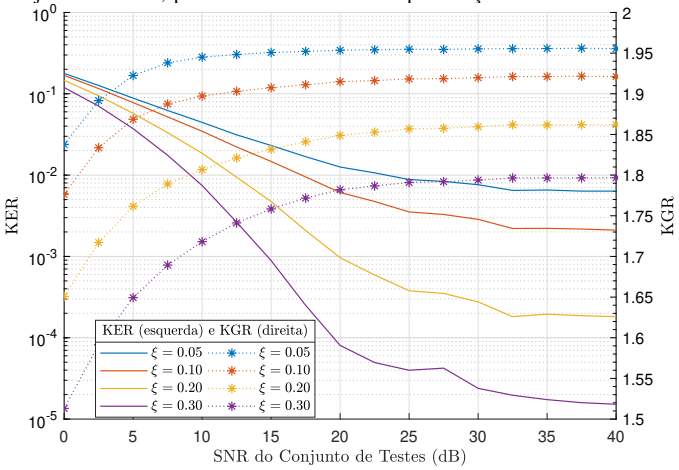
A Figura 4 ilustra a evolução da função de perda ao longo das épocas para diferentes tamanhos do conjunto original de dados durante uma rodada de treinamento. Nesse contexto, mantém-se uma proporção de 80% entre o tamanho do conjunto de treinamento e o conjunto original, conforme os resultados previamente mencionados. O que se observa nessa análise é uma constatação que pode parecer contra-intuitiva à primeira vista. Embora a expectativa fosse de que o aumento no tamanho do conjunto original levasse a uma diminuição na função de perda, esse padrão se verifica para conjuntos de dados originais de 20, 40, 60 e 80 mil usuários. Em outras palavras, quanto maior o conjunto de dados, menor a função de perda.

Contudo, uma anomalia surge quando o conjunto original contém 100 mil usuários, visto que a função de perda nesse caso se apresenta maior do que nos cenários com 60 e 80 mil usuários. Esse fenômeno aponta para a complexidade da questão, destacando que a ampliação indiscriminada do número de usuários não representa uma estratégia sempre eficaz para melhorar a acurácia da rede neural. A razão para isso reside no problema do viés dos dados. À medida que a quantidade de usuários aumenta, a rede neural passa a considerar peculiaridades específicas entre os canais dos usuários, que na realidade devem ocorrer raramente e serem insignificantes na generalização.

Contudo, com o aumento do número de usuários, esses cenários peculiares acabam por acontecer com maior frequência, levando a rede neural a atribuir um peso muito maior que o necessário para esses casos raros. Consequentemente, o processo de convergência torna-se mais demorado e a capacidade de generalização da rede é prejudicada. A análise da figura sugere que o tamanho ideal para o conjunto original situa-se em torno de 60 a 80 mil usuários, faixa na qual a rede neural demonstra melhor desempenho em termos de acurácia e generalização.

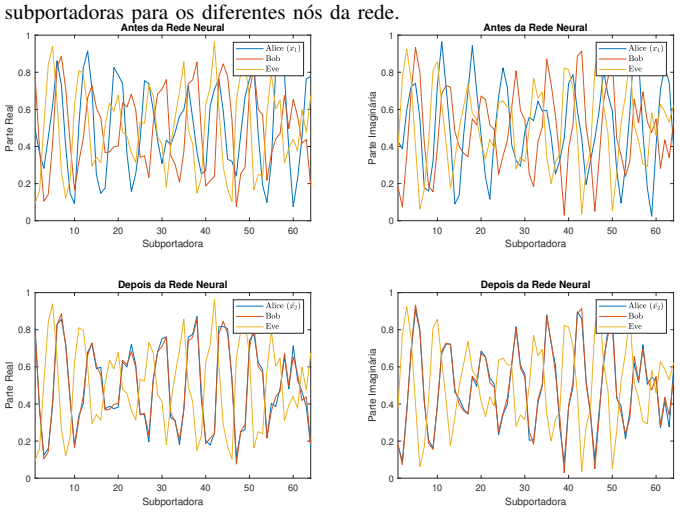
Uma vez analisada a performance da rede neural, os parâmetros obtidos através das análises acima foram utilizados para a construção da Figura 5, que explicita, para diferentes fatores de quantização, ξ, o KER e o KGR em função da relação sinal-ruído (SNR, do inglês *signal-to-noise-ratio*)

Figura 5. Variação nas métricas de KER e KGR em função da SNR do conjunto de testes, para diferentes fatores de quantização.



Considerando um conjunto de 60 mil usuários, em que 80% desse conjunto é destinado ao treinamento da rede neural sem ruído. Nesta etapa foi considerado o nível de quantização $K = 2$. Fonte: Autor.

Figura 6. Visualização dos elementos dos vetores de coeficiente de canal nas subportadoras para os diferentes nós da rede.



Considerando um conjunto de 60 mil usuários, em que 80% desse conjunto é destinado ao treinamento da rede neural sem ruído. Fonte: Autor.

do conjunto de testes. A SNR é uma medida que indica a proporção entre a potência do sinal desejado (sinal) e a potência do ruído presente em um sistema de comunicação ou em qualquer sinal elétrico ou eletrônico. Em geral, quanto maior o valor da SNR, melhor é a qualidade do sinal em relação ao ruído, o que significa que o sinal é mais claro e menos distorcido pelo ruído [23].

Os conjuntos de dados disponibilizados pelo DeepMIMO não incluem ruído intrínseco nos canais. No entanto, considerando a aplicação prática de sistemas de telecomunicação, é realista supor que os canais podem conter algum nível de ruído. Portanto, é necessário que a rede neural possua generalidade suficiente para lidar com essas questões práticas, mesmo quando treinada com um conjunto de dados teórico. Para simular essa situação mais próxima da realidade, os coeficientes de canal do conjunto de testes foram degradados antes da etapa de pré-processamento através da adição de ruído Gaussiano branco aditivo (AWGN, do inglês *additive white Gaussian noise*). Essa adição de ruído é crucial para examinar o desempenho da geração de chave aleatória de forma mais realista, assemelhando-se mais ao que aconteceria em uma situação prática.

Inicialmente, a Figura 5 ilustra o efeito do fator de quantização nas métricas de KER e KGR, cujos valores ideais são baixos e próximos de 2, respectivamente. Observa-se que à medida que o valor de ξ aumenta, a métrica KER diminui, uma vez que a ampliação da zona de guarda resulta em menos oportunidades de erro entre os bits das chaves geradas em Alice e Bob. No entanto, simultaneamente, a métrica KGR se afasta de 2, porque com uma zona de guarda mais ampla, a probabilidade de valores analógicos nesta faixa serem codificados como -1 aumenta consideravelmente.

Consequentemente, a escolha do valor do fator de quantização ξ torna-se altamente dependente da aplicação específica para a qual a rede neural será empregada. Além disso, a figura sugere uma relação proporcional entre o aumento de ξ e as métricas KER e KGR, com um impacto mais acentuado na KER. Portanto, em muitos cenários em que a privacidade é crucial para o sistema de comunicação, a opção por um KGR ligeiramente menor pode ser vantajosa para reforçar a segurança do sistema.

Além disso, a figura também ilustra um resultado previsível: quanto mais nítido for o sinal no conjunto de testes, ou seja, quanto maior for a SNR do conjunto de testes, maior será a capacidade da rede neural em realizar previsões precisas, especialmente quando treinada com dados isentos de ruído. No entanto, é importante destacar que essa melhoria não segue uma tendência linear. A introdução do ruído revela o impacto das funções de ativação e das camadas na própria rede neural. A rede aplica uma sequência de transformações não lineares para minimizar a função de perda, e isso pode resultar em comportamentos complexos que não seguem um padrão simples de melhoria constante com o aumento da SNR.

Por fim, a Figura 6 ilustra a comparação entre os valores dos coeficientes de canal normalizados obtidos por Alice, Bob e Eve, separados nas partes real e imaginária. A figura claramente demonstra o efeito da rede neural, que, a partir dos valores do vetor $\mathbf{x_1}$, estima um vetor $\hat{\mathbf{x_2}}$ extremamente próximo ao vetor obtido por Bob.

Por meio da figura, é possível observar que o canal espião apresenta uma diferença significativa em relação aos canais legítimos. Além disso, devido à distância que o separa, superior à metade do comprimento de onda das frequências de comunicação, não existe qualquer tipo de correlação com os canais legítimos. Somente através da utilização da rede neural, Alice é capaz de aproximar o canal de Bob.

Esta figura oferece uma visualização importante das capacidades da rede neural em inferir informações precisas sobre os coeficientes de canal, demonstrando a eficácia do processo de geração de chaves aleatórias proposto.

VIII. Conclusão

Ao longo deste trabalho, foi explorada de maneira abrangente a aplicação de redes neurais na geração de chaves aleatórias na camada física de sistemas de comunicação sem fio, abordando desde a aquisição de conjuntos de treinamento e teste até a análise de métricas de desempenho essenciais para avaliar a confiabilidade e segurança do sistema proposto.

No decorrer das análises, identificou-se que a seleção adequada da taxa de aprendizagem é um aspecto crucial para o sucesso da rede neural. Observou-se que taxas muito baixas podem levar a uma convergência lenta, enquanto taxas muito altas podem resultar em dificuldades de convergência e presença de mínimos locais. Por meio de simulações, foram encontrados valores ideais de taxa de aprendizagem que permitiram um equilíbrio entre rapidez de convergência e estabilidade, facilitando a generalização dos resultados.

Além disso, as simulações revelaram a importância da quantidade de dados de treinamento na eficácia da rede neural. Contrariamente à intuição inicial, identificou-se um ponto de saturação em que o aumento excessivo do tamanho do conjunto de treinamento não resulta em ganhos significativos de desempenho, devido à tendência da rede neural em capturar particularidades de casos raros e não generalizar adequadamente.

Ao introduzir ruído nos conjuntos de teste, simulando um cenário mais próximo da aplicação prática, foi possível avaliar a resiliência da rede neural em face de condições adversas. Os resultados mostraram a capacidade da rede em lidar com variações realísticas de sinal-ruído, corroborando sua robustez e adequação para ambientes reais de comunicação.

Por fim, foi demonstrado que a rede neural é capaz de estimar com precisão os coeficientes de canal, gerando uma chave aleatória próxima à obtida por Bob. Isso valida a viabilidade e eficácia do método proposto para a geração segura de chaves na camada física.

Em suma, este trabalho contribui com insights interessantes sobre a aplicação de redes neurais na geração de chaves aleatórias, destacando aspectos cruciais para seu funcionamento eficaz. Compreende-se que a otimização desses parâmetros e a avaliação adequada das métricas de desempenho são essenciais para garantir a segurança e confiabilidade dos sistemas de comunicação sem fio baseados nesse método.

Referências

[1] SHAH, A. F. M. S. A survey from 1g to 5g including the advent of 6g: Architectures, multiple access techniques, and emerging technologies. In: *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. [S.l.: s.n.], 2022. p. 1117–1123.

[2] TATARIA, H. et al. 6g wireless systems: Vision, requirements, challenges, insights, and opportunities. *Proceedings of the IEEE*, v. 109, n. 7, p. 1166–1199, 2021.

[3] PORAMBAGE, P.; LIYANAGE, M. *6G security and privacy: A comprehensive guide*. Hoboken, NJ: Wiley-Blackwell, 2023.

[4] ZENG, K. Physical layer key generation in wireless networks: challenges and opportunities. *IEEE Communications Magazine*, v. 53, n. 6, p. 33–39, 2015.

[5] JIAO, L. et al. Physical layer key generation in 5g wireless networks. *IEEE Wireless Communications*, v. 26, n. 5, p. 48–54, 2019.

[6] REN, K.; SU, H.; WANG, Q. Secret key generation exploiting channel characteristics in wireless communications. *IEEE Wireless Communications*, v. 18, n. 4, p. 6–12, 2011.

[7] MITEV, M. et al. What physical layer security can do for 6g security. *IEEE Open Journal of Vehicular Technology*, v. 4, p. 375–388, 2023.

[8] LI, G. et al. Physical layer key generation in 5g and beyond wireless communications: Challenges and opportunities. *Entropy*, v. 21, n. 5, 2019. ISSN 1099-4300. Disponível em: <https://www.mdpi.com/1099-4300/21/5/497>.

[9] SIRIWARDHANA, Y. et al. Ai and 6g security: Opportunities and challenges. In: *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. [S.l.: s.n.], 2021. p. 616–621.

[10] ZHOU, Z.-H. *Machine learning*. [S.l.]: Springer Nature, 2021.

[11] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

[12] WANG, W. et al. A wireless secret key generation method based on chinese remainder theorem in fdd systems. *Science China Information Sciences*, v. 55, 07 2012.

[13] LIU, B.; HU, A.; LI, G. Secret key generation scheme based on the channel covariance matrix eigenvalues in fdd systems. *IEEE Communications Letters*, v. 23, n. 9, p. 1493–1496, 2019.

[14] WU, X. et al. A secret key generation method based on csi in ofdm-fdd system. In: *2013 IEEE Globecom Workshops (GC Wkshps)*. [S.l.: s.n.], 2013. p. 1297–1302.

[15] PENG, L. et al. An investigation of using loop-back mechanism for channel reciprocity enhancement in secret key generation. *IEEE Transactions on Mobile Computing*, v. 18, n. 3, p. 507–519, 2019.

[16] LI, G. et al. Constructing reciprocal channel coefficients for secret key generation in fdd systems. *IEEE Communications Letters*, v. 22, n. 12, p. 2487–2490, 2018.

[17] ZHANG, X. et al. Deep-learning-based physical-layer secret key generation for fdd systems. *IEEE Internet of Things Journal*, v. 9, n. 8, p. 6081–6094, 2022.

[18] LI, G. et al. High-agreement uncorrelated secret key generation based on principal component analysis preprocessing. *IEEE Transactions on Communications*, v. 66, n. 7, p. 3022–3034, 2018.

[19] LI, G. et al. Encrypting wireless communications on the fly using one-time pad and key generation. *IEEE Internet of Things Journal*, v. 8, n. 1, p. 357–369, 2021.

[20] SILVA, G. M. *KGNet: Key Generation Network*. 2023. Disponível em: <https://github.com/gus7avoms7/KGNet>.

[21] KINGMA, D.; BA, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[22] ALKHATEEB, A. *DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications*. 2019.

[23] MEDEIROS, J. D. O. *Princípios De Telecomunicações: TEORIA E PRÁTICA*. ERICA, 2016. ISBN 9788536516288. Disponível em: <https://books.google.com.br/books?id=1rdwmgEACAAJ>.