Accuracy-Aware Compression of Channel Impulse Responses using Deep Learning

Conference Paper · December 2021

6 authors, including:

Sebastian Kram
Fraunhofer Institute for Integrated Circuits IIS
16 PUBLICATIONS **75** CITATIONS
SEE PROFILE

Maximilian Stahlke
Technische Hochschule Nürnberg Georg Simon Ohm
6 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Tobias Feigl
Friedrich-Alexander-University of Erlangen-Nürnberg
30 PUBLICATIONS **167** CITATIONS
SEE PROFILE

Christopher Mutschler
Fraunhofer Institute for Integrated Circuits IIS
60 PUBLICATIONS **476** CITATIONS
SEE PROFILE

2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 29 Nov. – 2 Dec. 2021, Lloret de Mar, Spain

Accuracy-Aware Compression of Channel Impulse Responses using Deep Learning

Thomas Altstidl*†, Sebastian Kram‡*, Oskar Herrmann*, Maximilian Stahlke*, Tobias Feigl*, and Christopher Mutschler*

{ thomas.r.altstidl | sebastian.k.kram | oskar.herrmann }@fau.de
{ maximilian.stahlke | tobias.feigl | christopher.mutschler }@iis.fraunhofer.de

*Fraunhofer Institute for Integrated Circuits IIS
Precise Positioning and Analytics Department
Nürnberg, Germany

†Machine Learning and Data Analytics Lab
‡Institute of Information Technology
Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Germany

***Abstract*—Ultra-wideband (UWB) systems based on Channel State Information (CSI) estimate the position of mobile nodes within an environment by using Channel Impulse Responses (CIRs) of multiple stationary nodes. These contain spatial information caused by environment interactions such as reflections and scattering. To estimate positions from CSI of stationary nodes, we must transmit them to a centralized node. This introduces considerable communication overhead.**

We present a large-scale study to determine whether CSI can be compressed into a small set of underlying latent variables that describe the most valuable information. We evaluate multiple neural network architectures containing encoding (compressing) and decoding (reconstructing) components and compare them to the state-of-the-art compression techniques Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). We show that fully connected autoencoders achieve the lowest error, outperforming both DCT and DWT. Further experiments prove that the reconstructed CSI can be used for positioning with only mild performance deterioration at a compression of $>$97% and even when trained on a different environment.

***Index Terms*—Compression, Channel Impulse Response, Channel State Information, DCT, DWT, (Variational) Autoencoder.**

I. Introduction

Precise radio frequency (RF) indoor positioning systems (IPS) are key to many location-based services such as indoor navigation and object tracking [1]. Recently, Ultra-wideband (UWB) gained a lot of attention as an accurate and resilient, yet cost-efficient technology [2], [3]. While under ideal line-of-sight (LOS) conditions UWB ranging may yield sub-centimeter accuracy [2], non-line-of-sight (NLOS) propagation caused by obstructions and reflections is still challenging.

Most UWB position estimators use parameters such as the time of arrival (TOA), the time difference of arrival (TDOA), the received signal strength (RSS), the angle of arrival (AOA), or combinations [2]. These parameters are extracted from the Channel State Information (CSI) obtained from the radio signal that is sent from the transmitter to the receiver(s). Thus, the entire received signal is reduced to a single value used in a downstream task, e.g., to estimate the position of the transmitter. This compression is, of course, accompanied by a loss of information [4]. Yet Channel Impulse Responses (CIRs), as a time series of correlation values between the transmitted and received signal, describe complex signal propagation, like reflection, absorption, scattering, and diffraction of the signal. Fig. 1 shows one in LOS and one in NLOS conditions.

Recently, there is a lot of interest in exploiting the full potential of these CIRs. This includes models of the CIR [5] and the channel characteristics specific to an environment [6]–[8]. Other research focuses on extracting multipath components (MPCs) [7], [9] to extract additional spatial information [10], [11]. The same information is implicitly used for mitigating NLOS propagation conditions [12], i.e., for CIR- or CSI-based fingerprinting [13], [14] or LOS/NLOS classification [15].

However, as CIRs contain detailed correlation information, using them in positioning-related tasks not only requires high computational resources, but also centralized processing of all CIRs observed within a network. This introduces a significant communication load. Hence, it is desirable to compress CIRs at the receiving units into a set of informative features that still keep the required spatial information. While well-known general-purpose signal compression methods such as the dis-

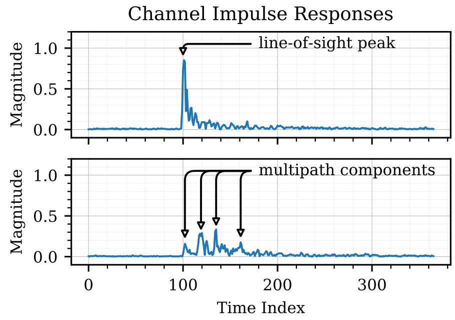

Fig. 1. Two samples of channel impulse responses (CIRs): line-of-sight propagation (top) and multipath propagation with prominent multipath components, primarily attributable to reflections (bottom). The initial peak when the signal arrives is labelled. Both signals are subject to diffuse MPCs and noise.

crete cosine transform (DCT) [16] or the discrete wavelet transform (DWT) [17] can also be applied, their compression performance is limited, as they do not include information on the variety of position-related information that is contained sparsely within the CSI [18]. To address these problems, recent work has highlighted the usefulness of autoencoder-based compression [19], [20]. Although these are often optimal, there is little research specifically targeting CIRs [21]. Many crucial aspects, such as the choice of autoencoder type and architecture and its applicability to real-world IPS tasks, are thus still unexplored.

To close this gap of previous work [19], [21], we propose a full autoencoder study for CIRs and a real-world evaluation. Our three main contributions are thus the following:

- We present a study of CIR compression using autoencoder types with fully connected and convolutional layers, both in their vanilla and variational forms. In addition, we benchmark against state-of-the-art DCT and DWT.
- We present an initial study on generalization characteristics across different environments. We show that our compression model generalizes well to previously unknown environments and still provides significant compression.
- We evaluate the performance of fingerprinting methods using both compressed representations and reconstructed signals, obtaining a positioning error of $0.59\ \mathrm{m}$ using the reconstructed signals (compared to $0.42\ \mathrm{m}$ for the original signals), even when generalizing across environments.

The remainder of this paper is structured as follows. Sec. II discusses related work. Sec. III introduces our localization system and the collected dataset. The architectures for compression and fingerprinting are described in Sec. IV. Finally, Sec. V discusses experimental results before Sec. VI concludes.

II. Related Work

Compression is a broad field that is primarily driven by the need to reduce both memory and bandwidth, and the requirements and methods are highly application specific [22]. For tasks where slightly deviating reconstructions are acceptable, lossy techniques achieve the highest compression rates [22]. Since radio-based localization yields exact positions on a relatively small parameter set such as TOA, it is sufficient to find lossy compression methods that preserve the essential features. That is why we only discuss such methods.

The DCT interprets signals as the sum of cosine functions of varying frequencies. DCT is well known from its applications on images [23], videos [24], and audio [25] and biomedical signals [20], [26]. However, in the domain of radio signals DCT shows poor performance [16], [19] (see also our experimental results in Sec. V). The DWT is similar to DCT but decomposes the signals into a set of wavelet functions. DWT has been applied to images [27] and biomedical signals [20], [28]. It has also been proposed for compression of CSI [17], which helps for downstream tasks such as keystroke recognition [29], [30]. However, similar to DCT, the compression performance of DWT is limited (see Sec. V). Other approaches such as the Karhunen-Loeve transform (KLT) [19], Principal Component Analysis (PCA) [19], [20], Lightweight Temporal Compression (LTC) [20], [26] and Symbolic Aggregate Approximation (SAX) [26] are also popular compression methods. However, these are either computationally intensive or outperformed by deep learning [19] and autoencoders [20].

Recently, deep learning approaches have gained wide-spread attention for compression tasks. For instance, autoencoders achieve improvements in areas dominated by traditional compression algorithms, such as images [31] and videos [32]. Both vanilla autoencoders [33] and β-variational autoencoders [34] have also shown favorable results for compression and feature extraction in biomedical signals. It has also gathered significant attention in the field of CSI compression to reduce overhead. Approaches include fully connected autoencoders [35], [36], encoding and decoding convolutional neural networks (CNNs) [37], [38], an encoding CNN coupled with a decoding recurrent neural network (RNN) [19] and variational autoencoders [39], [40]. A study of three convolutional autoencoders on UWB CIR denoising suggests large kernels and shallow networks are preferred [41]. In terms of comparative studies, Del Testa et al. [20] compared autoencoder compression to DCT, DWT, PCA, and LTC and found these to outperform the classical methods on biomedical signals. Similar results were obtained by Liao et al. [19], with encoder-decoder schemes outperforming DCT, KLT, and PCA. Among the traditional methods, DCT performs worse than DWT [20], while PCA or KLT generally give the best results [19], [20].

To the best of our knowledge, there are only a few data-driven compression methods for channel impulse responses for the specific task of localization or fingerprinting. Nerguizian et al. [42] used a DWT for fingerprinting with 13% loss in positioning accuracy. Tsai et al. [36] show that autoencoders have negligible effects on fingerprinting using location databases. Fontaine et al. [21] show a 29% higher accuracy after correcting for multipath effects using compressed CIR features derived from autoencoders. However, there is no preliminary work that reports the impact of the proposed architecture on the compromise between compression and reconstruction accuracy, and they all leave unclear how well their methods generalize to unknown environments.

In conclusion, although both DCT and DWT are still used successfully in multimedia applications, related work suggests that autoencoders offer the best combination of compression ratio and reconstruction error when a sufficiently large training dataset is available. The few previous works on CIR compression show similar findings [19], [20]. Our approach builds on the existing knowledge. In contrast, we concentrate on deriving an optimal architecture that offers high compression rates and low reconstruction errors and can thereby generalize between trained and unknown propagation environments. As a result, our overall compression pipeline requires little to no fine-tuning for final deployment in different environments.

III. Background and Datasets

Multipath propagation occurring in indoor environments can significantly decrease the quality of IPS. Fig. 1 exemplifies

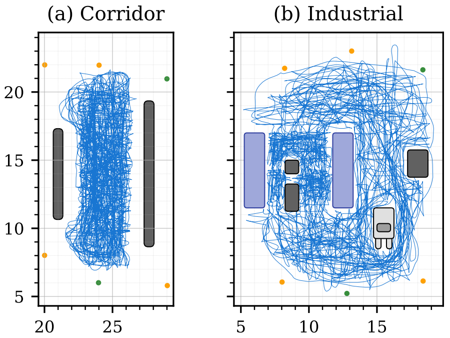

Fig. 2. Visualization of signal collection paths in dense multipath environments: Receivers and objects in (a) Corridor scenario and (b) Industrial scenario. Object locations are approximate and for illustrative purposes only. Orange anchors are used for training, green anchors are used for validation.

the magnitude of two sample CIRs. The top row shows a communication link with a clear LOS component (the peak is marked). The bottom row shows a link with significant multipath effects and without distinguishable LOS path (we marked the multiple distinct peaks produced by specular reflection). In real-world environments, such as industrial, indoor or urban scenarios, highly complex propagation conditions lead to such heavy multipath propagation. Thus, the resulting CIRs contain spatial information that is difficult to model explicitly.

To learn a model that can compress and reconstruct the multitude of possible signals, it is important to collect a diverse and representative database of CIRs. Towards this end we use the dataset published in [8]. We employ the Decawave DW1000 platform and use a total of six anchors configured as receivers at known and fixed positions along with one moving transmitter that periodically sends signal bursts that are received by the anchors. Each of the recorded CIRs contains a total of $n_c = 366$ samples. Reference positions are determined using a Nikon iGPS optical reference system with accuracy specified in the millimeter range.

The dataset consists of three different scenarios collected in a 30m × 40m area. The *Clean Scenario* contains no obstacles and thus is mostly free of significant reflections. Nevertheless, any suitable system must also cope with this baseline. The *Corridor Scenario* contains two reflective and absorptive walls that introduce multipath propagation. The location of the walls relative to the anchors and transmitter positions is illustrated in Fig. 2. The reflective sides of these walls point inwards. The last scenario, referred to as the *Industrial Scenario*, contains more diverse objects such as shelves, boxes, and a forklift. It is the most complex and most representative scenario. The dataset holds a total of $819,234$ CIRs, with $198,460$ CIRs for the *Clean Scenario*, $318,827$ CIRs for the *Corridor Scenario*, and $301,947$ CIRs for the *Industrial Scenario*.

Our main study uses all scenarios, with orange anchors for training and green for early stopping and model selection (Fig. 2). We retrain on the *Industrial Scenario* with a random training and validation split, then evaluate generalization on the *Corridor Scenario*. Finally, the fingerprinting application study is performed only on the *Corridor Scenario*.

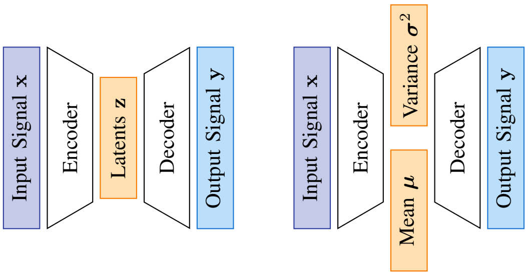

Fig. 3. Structure of autoencoder networks. The input $\mathbf{x}$ is first compressed to latent variables $\mathbf{z}$ and then reconstructed as output $\mathbf{y}$. For the variational type the latent space are normal distributions with mean μ and variance σ^2 from which the subsequent decoder samples.

IV. Method

We train autoencoders to efficiently compress the CIR signals while still retaining the spatial information embedded in them (i.e., LOS peaks and MPCs). A compression directly at the receiving nodes reduces data transmission between the nodes and a centralized position server. This is essential for high-frequency real-time positioning. Having a smaller set of informative and semantic features should also help to reduce the amount of required training data for downstream tasks such as fingerprinting. This section describes four (prominent) autoencoder types that form the basis of our comprehensive study along with their hyperparameters.

Autoencoders use the two components as illustrated in Fig. 3: an encoder $f(\mathbf{x})$ maps the input signal into a latent representation $\mathbf{z}$ while a decoder $g(\mathbf{z})$ uses $\mathbf{z}$ to reconstruct the input into $\mathbf{y}$. The information loss of the composite $g(f(\mathbf{x}))$ is minimized. Variational autoencoders (Fig. 3, right) represent the latent space using normal distributions and learn a set of means $\boldsymbol{\mu}$ and variances $\boldsymbol{\sigma}^2$ to encourage a more semantic and disentangled representation in the compressed latent space. Each input z_i for the subsequent layer is then randomly sampled from the respective normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ [43]. It thus formulates the decoder as a generative stochastic process that yields signals reflecting the true posterior distribution.

There are two main layer types of interest: the fully connected (FCN) layer and the convolutional (CNN) layer. The FCN layer applies linear transformations to the input, which does not scale well with input size in terms of computations. The CNN-layer reduces this complexity by computing a cross-correlation on each small neighborhood separately, albeit with shared weights. This is suitable for the given data, as time series inherently exhibit temporal dependencies.

Taking into account the types of autoencoders and layers, we group our models into four different categories: FCN-AE (vanilla autoencoder with FCN layers), CNN-AE (vanilla autoencoder with CNN layers), FCN-VAE (variational autoencoder with FCN layers), and CNN-VAE (variational autoencoder with CNN layers). The investigated network structures are shown in Fig. 4. We optimized the hyperparameters through an extensive randomized search, see Table I.[1]

[1] A random search performs as well as grid search in many scenarios [44], and was chosen for practicality due to the large number of hyperparameters.

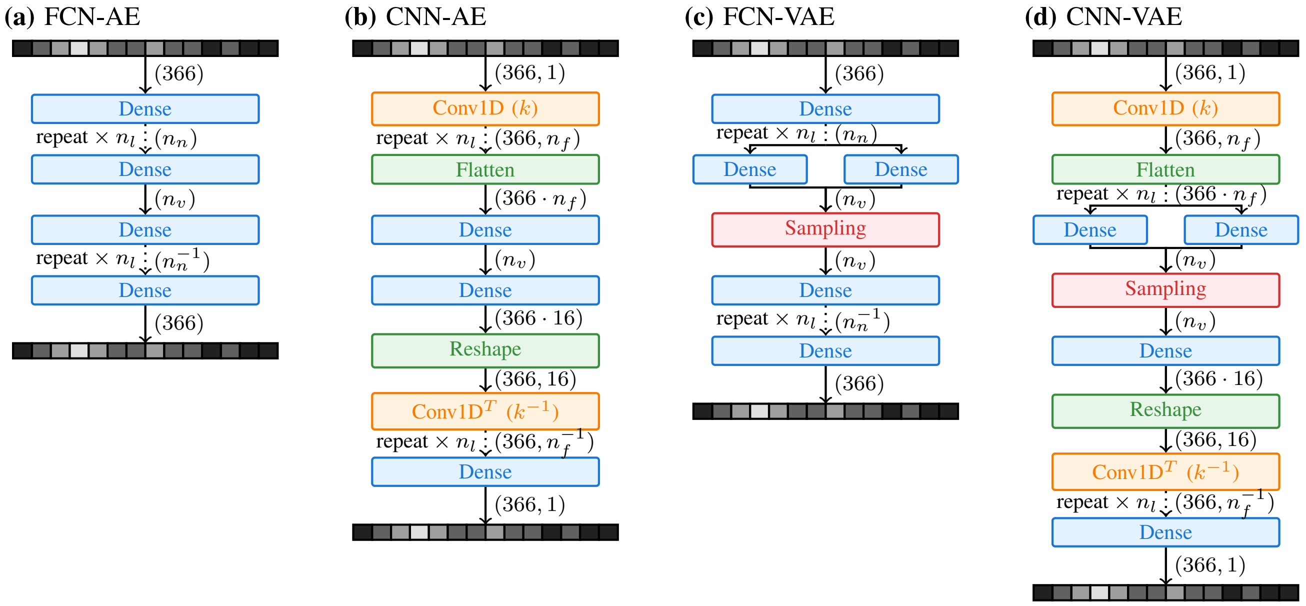

Fig. 4. Illustration of neural network types used throughout our study. For a detailed description of the models and hyperparameters, please refer to the text. Dense refers to a fully connected layer, Conv1D to a convolutional layer and Conv1D^T to a transposed convolutional layer. Flatten and Reshape are used to modify the array shapes, which are denoted by the numbers in parentheses throughout the network. Sampling is a special layer used only for VAE models that randomly samples from the latent distribution. Some layers are repeated n_l times, in which case each has its own parameters taken from a list of n_l values. For the decoder, the order of the layers is reversed compared to the encoder, which is denoted by the $^{-1}$ superscript.

TABLE I
HYPERPARAMETER SEARCH SPACE.

Parameter	Distribution	Values
n_l	uniform	0, …, 4
n_v	uniform	1, …, 8
n_n	uniform	1, …, 366
n_f	uniform	1, …, 64
k	uniform	1, …, 17
act	uniform	relu, elu, sigmoid, hard_sigmoid
n_b	uniform	1, …, 512
opt	uniform	sgd, rmsprop, adadelta, adam, nadam
lr	log. uniform	10^{-5}, …, 10^{-1}
β	log. uniform	10^{-7}, …, 10^{1}

n_l: # FCN/CNN layers between input and latent space, and latent space and output; n_v: # latent variables (for AE: # neurons in latent layer; for VAE: # means and variances); n_n: # hidden units per layer (only FCN); n_f: # convolutional kernels per layer (only CNN). k: filter size of kernel per layer (only CNN); act: activation following each layer; n_b: batch size used for training. opt: optimizer used for training; lr: learning rate of optimizer; β: β-factor for KL divergence of loss term (only VAE)

Let us now explain how we approach the optimization criteria to train the various autoencoders. To ease the subsequent mathematical formulations, we now define a CIR $\mathbf{c}$ as a sequence of values c_i, where i is a time-based index and n_c is the number of correlation values, such that the entire signal $\mathbf{c} = (c_1, c_2, \ldots, c_{n_c-1}, c_{n_c})$.

During loss calculations, we require a measure comparing the original CIR $\mathbf{c}^{(1)}$ with the reconstructed CIR $\mathbf{c}^{(2)}$. For our case, we chose the well-established mean squared error (MSE) for the loss, the computation of which is formalized as:

$$\mathcal{L}_{\mathrm{AE}}(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) = \frac{1}{n_c} \cdot \sum_{i=1}^{n_c} (c_i^{(1)} - c_i^{(2)})^2. \tag{1}$$

In the case of a variational autoencoder the Kullback-Leibler (KL) divergence between the individual latent distributions $\mathcal{N}(\mu_i, \sigma_i^2)$ and the standard normal distribution $\mathcal{N}(0, 1)$ is added. This loss term can be formalized as:

$$\mathrm{KL}(\mu, \sigma) = \frac{1}{2} \sum_{i=1}^{n_v} (\mu_i^2 + \sigma_i^2 - \log \mu_i^2 - 1). \tag{2}$$

Subsequent layers of the network can suitably transform the distribution $\mathcal{N}(0, 1)$, which makes it the default choice for variational autoencoders even though others are possible [43].

The complete loss for a variational autoencoder is then computed by adding this term to the loss for vanilla autoencoders defined in Eqn. 1, with β acting as an additional parameter for balancing the two different losses [45]:

$$\mathcal{L}_{\mathrm{VAE}}(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \mu, \sigma) = \mathcal{L}_{\mathrm{AE}}(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) + \beta \cdot \mathrm{KL}(\mu, \sigma). \tag{3}$$

Model training involves the optimization of these losses. One standard technique is mini-batch stochastic gradient descent (SGD) [46]. More advanced techniques such as RMSProp, AdaDelta, Adam and Nadam have also been proposed [47]. As none of these have been shown to be generally superior, they form a part of our study.

We split our data into training and validation sets along anchors. All three scenarios (*Clean*, *Corridor*, and *Industrial*) are included in the data. For training, we use the orange anchors shown in Fig. 2 and for validation we use the remaining two green anchors. This ensures that our validation data is different w.r.t. effects such as multipath propagation. The validation loss is computed after each epoch of training.

We train a total of 9,142 configurations based on the parameters from Table I. A full grid search is infeasible considering

TABLE II
HYPERPARAMETER STUDY RESULTS

Model	Lat.	Layers[1]	Batch	Optimizer[2]	Beta	Score	#Params
FCN-AE	8	343 – relu – 110 – sigm. – 290 – relu	127	Nadam($1.06 \cdot 10^{-3}$)	—	$2.54 \cdot 10^{-4}$	396,836
CNN-AE	7	27×17 – elu – 27×13 – relu – 16×5 – elu	486	Adam($1.99 \cdot 10^{-3}$)	—	$2.82 \cdot 10^{-4}$	119,400
CNN-VAE	8	54×12 – relu – 36×13 – elu	226	Adam($3.56 \cdot 10^{-4}$)	$4.95 \cdot 10^{-7}$	$2.83 \cdot 10^{-4}$	320,507
FCN-VAE	8	97 – sigm. – 308 – hard sigm. – 174 – elu	98	Nadam($2.04 \cdot 10^{-3}$)	$7.07 \cdot 10^{-6}$	$3.22 \cdot 10^{-4}$	243,656

[1] Describes # hidden units (for FCN) or # convolutional kernels × filter size (for CNN), plus the activation. Decoder layers mirror the encoder layers.
[2] Number in parentheses is learning rate.

the number of parameters involved. We stop training when the validation loss does not improve for five epochs, for a maximum of 100 epochs.

One of the primary challenges of IPS are the complex propagation conditions in obstructed environments. One way to deal with this is fingerprinting. This environment-specific method works by compiling a database of known signals at known locations that can be queried to estimate the current position, e.g., using deep learning [14], [48]. This state-of-the-art mitigation method is further evaluated to estimate the impact of compression.

V. RESULTS

First, we discuss our hyperparameter study (Sec. V-A). Next, we comparatively evaluate the compression (Sec. V-B) and analyze the impact on fingerprinting (Sec. V-C).

A. Hyperparameter Study

Out of the total 9,142 trained configurations we discard 6,084 models as they did not achieve reasonable reconstruction results on the validation data and such are unsuitable in practice.[2] The results indicate that the more important design considerations is the network type. Towards this end, Fig. 5 shows a kernel density plot for the MSE scores on the validation data, split by network type. In addition, Table II lists the best model within each group of network types. Subsequently, we discuss some of our key findings.

Dense models are better, but less consistent than convolutional models. This suggests that convolutional models are the more natural fit for the time-correlated CIR data. Still, among the top ten models, eight are of the FCN-AE type, indicating that properly configured variants of dense models outperform other types. As dense networks connect to all input nodes (unlike convolutional networks) they are capable of capturing more complex correlations within the CIR signal. However, CNN models seem to be less sensitive to hyperparameters, which makes them easier to be tuned in practice.

Vanilla models outperform variational models. Only a single variational model, CNN-VAE, is among the top 100 models, suggesting that these are only partially applicable to the given domain. One possible explanation is the conflicting goal of the variational autoencoder, which encourages standard normal distributions in the latent space, thus trading some of the compression effectiveness. While we assumed that modelling of the input distribution could lead to a better generalization, we could not see this in our evaluation. Therefore we do not consider VAEs in our further discussion.

RMSProp, Adam, and Nadam optimizer equally yield best performances. SGD and Adadelta, on the other hand, perform significantly worse as they are more sensitive to the selected learning rate [49]. This is in line with other published research [50]. Learning rates between 10^{-4} and 10^{-3} produce

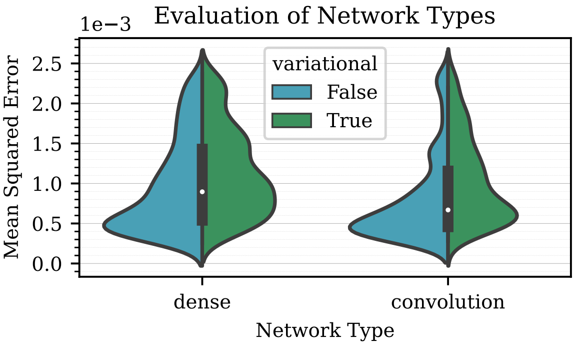

Fig. 5. Kernel density plot of MSE scores on validation data, split by neural network type. Models with scores lower than $2.37 \cdot 10^{-3}$ were discarded.

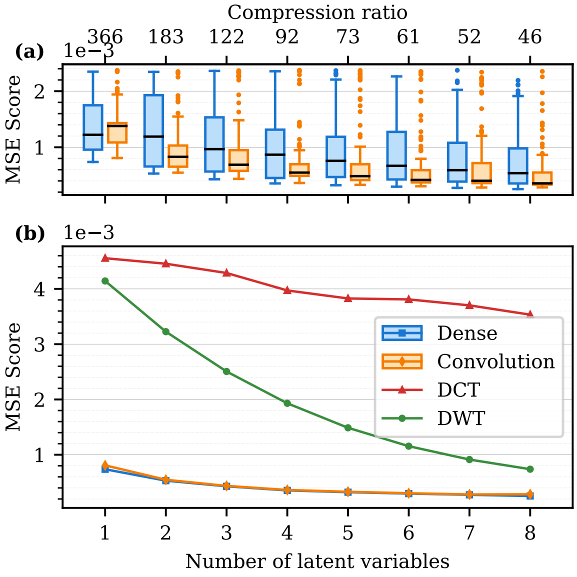

Fig. 6. MSE scores by number of latent variables n_v (bottom axis) and compression ratio (top axis). **(a)** Box plots of FCN-AE and CNN-AE study models, where black lines visualize medians, colored bodies quartiles and colored whiskers non-outlier extrema. **(b)** Best FCN-AE and CNN-AE models along with DCT and DWT for reference. Best viewed in color.

[2] A model is considered reasonable if its error is lower than $2.37 \cdot 10^{-3}$, obtained when predicting the average CIR of the data.

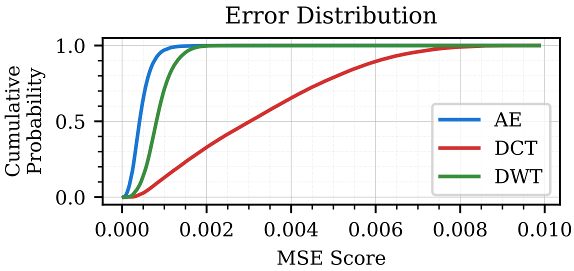

Fig. 7. Distribution of MSE scores for FCN-AE, DCT and DWT on testing data of *Corridor Scenario*. All three use $n_v = 8$ latent variables.

the best overall scores. We find that the batch size has no measurable impact on the overall results, suggesting the problem is simple enough that a few samples per batch suffice to guarantee convergence.

FCN-AE should use at least n_l=2 layers and performs best with certain activations. Early layers should use either ReLU or ELU, while later layers should use either sigmoid or hard sigmoid. One possible explanation is that the decoder benefits from the additional regularization caused by the limitation to the activation range $[0, 1]$. The individual number of neurons n_n per layer did not influence results much overall as long as a certain overall number of parameters was used.

CNN-AE should contain at least n_l=1 layer and again performs best with specific activations. The first layer should use either ReLU or hard sigmoid while all other layers should use the sigmoid activation. The layer count n_l is largely irrelevant besides the bare minimum of one layer. This is also in line with previous results [41]. Both filter count n_f and kernel size k do not show a significant impact on performance.

Roughly 10,000 parameters are needed to achieve reasonable results for both FCN-AE and CNN-AE. This makes it easy to deploy it on computationally restricted devices often encountered in IPS. Parameter counts in excess of around 50,000 do not significantly improve performance anymore.

The compression requires a trade-off between ratio and error. The achieved compression ratio is inversely related to the number of latent variables n_v. To be usable in practice, the compression needs to maintain suitably low reconstruction errors. Thus, Fig. 6 (top) shows the distribution of MSE scores for different n_v. As expected, the score follows an approximately exponential decay with respect to n_v. After $n_v = 4$, the relative improvements of each additional latent variable drop below 10% and might not be needed. Thus, $n_v = 4$ could be sufficient depending on the targeted application.

Summary: We recommend the usage of the FCN-AE type, with $n_l \geq 2$ layers and ReLU or ELU for the first layer and sigmoid or hard sigmoid for the later layers. For computing-constrained environments the parameters can be lowered to a few 10,000s by appropriately tuning the number of layers n_l and neurons per layer n_n. For bandwidth-constrained environments the number of variables can be reduced to $n_v = 4$, at the cost of additional reconstruction errors. While CNN-AE are less sensitive to hyperparameters, they lag behind FCN-AE when both are optimally configured.

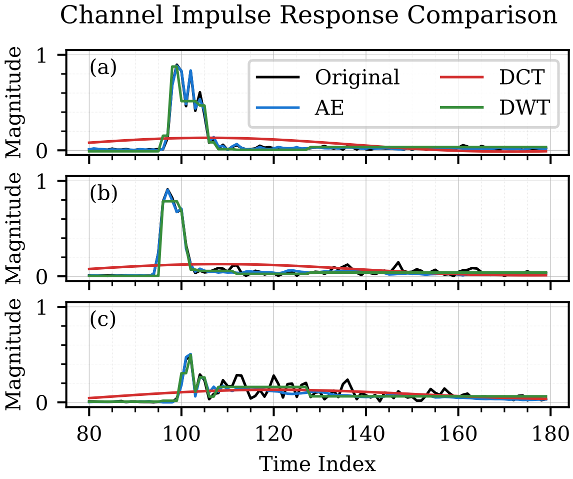

Fig. 8. Comparison of original with FCN-AE, DCT, and DWT reconstruction on three sample CIRs, cropped to $i \in [80, 179]$. Peak is well reconstructed for both FCN-AE and DWT, but missing for DCT. MSE scores for FCN-AE are (a) $0.81 \cdot 10^{-4}$, (b) $2.62 \cdot 10^{-4}$ and (c) $8.40 \cdot 10^{-4}$. FCN-AE uses $n_l = 3$ layers and $n_v = 8$ latent variables. First layer has $n_n = 343$ neurons with ReLU activation, second 110 with sigmoid and third 290 with ReLU.

B. *Baseline Comparison*

We also benchmark our autoencoders against two state-of-the-art compression methods, i.e., DCT (that keeps the n_v first components as features) and DWT (that keeps the n_v highest coefficients). Note that for DWT, an additional overhead is incurred as we need to store the index of the coefficient. Fig. 6 (bottom) shows the results of the best autoencoder models along with the baseline methods on the same validation data that we used for the hyperparameter study. While DWT outperforms DCT considerably, both cannot reach the level of compression performance that our best autoencoders yield. This is because both DCT and DWT are general-purpose algorithms that incorporate no domain knowledge, while the autoencoders implicitly learn and use the inherent characteristics of the data to achieve more efficient compression. he score distributions shown in Fig. 7 further highlights the superiority of our best FCN-AE model over DCT and DWT.

To enable both a better interpretation of the error values and a qualitative comparison of the methods, Fig. 8 shows three examples with reconstructions of FCN-AE, DCT, and DWT. The DCT does not capture the complexity of the signal, as it cannot describe the CIR structure with only a few cosine functions. The DWT reconstructs the signal more accurately, especially in the area of the peak, but still only provides a rough approximation of the position and shape of the peak. Our FCN-AE retains the exact shape of the important peak, while discarding the non-relevant noise later within the signal. This further highlights the capability of the autoencoder to select features by importance, as the peak is essential for TOA estimation and thus positioning.

C. *Real-World Applicability*

To assess the effects of compression in real-world applications, we evaluate a fingerprinting approach with deep learning

TABLE III
Generalization Study Results

Model	Validation MSE (Industrial Scenario)	Testing MSE (Corridor Scenario)	
FCN-AE	$1.95 \cdot 10^{-4}$	$4.12 \cdot 10^{-4}$	×2.11
CNN-AE	$2.32 \cdot 10^{-4}$	$4.90 \cdot 10^{-4}$	×2.11
FCN-VAE	$2.62 \cdot 10^{-4}$	$5.33 \cdot 10^{-4}$	×2.03
CNN-VAE	$2.38 \cdot 10^{-4}$	$4.79 \cdot 10^{-4}$	×2.01

on raw CIRs [14], [48]. Our proposed pipeline compresses the CIR at each receiver and then sends it to a local server for positioning. The central positioning neural network requires good reconstructions, since it makes use of important spatial information (e.g. reflections, diffraction and absorption).

We previously trained on all scenarios, which is not representative of a typical use case, where a trained model is deployed in a different environment. To simulate our outlined pipeline, we first retrain the best model of each category on the *Industrial Scenario* with a 80%/20% training and validation split and then evaluate on the *Corridor Scenario*.

Table III lists the resulting validation and testing scores. The FCN-AE model continues to perform best, while the FCN-VAE model performs worst. For all models, the errors are around twice as high in the testing scenario compared to the validation scenario. We subsequently select the FCN-AE compression model for further analysis. We then choose FCNs to estimate the location based on the six CIRs from each anchor. We evaluate three different scenarios, which train and test with the original CIRs, the reconstructed CIRs and the compressed features, respectively. The CIR-based networks (FCN-FP-1) compute two values representing the x- and y-coordinates based on the 6×366 CIR values. In a preliminary study, we found that the following parameters yield optimal results. The network contains three FCN layers using 800, 300, and 50 neurons, each followed by a ReLU activation. The feature-based network (FCN-FP-2) computes the coordinates using the 6×8 latent variables. This network uses three FCN layers with 30, 20, and 10 neurons, each followed by ReLU.

Training of the fingerprinting networks is done on the *Corridor Scenario*. The individual sets of six CIRs are sorted by timestamp, ensuring that the splits contain slightly different data within the same general area. We use 60% at the start for training purposes, with the next 20% kept for validation and the final 20% for testing. The evaluation criterion is the mean Euclidean distance between the predicted points on the testing data and the ground truth points obtained with a Nikon iGPS optical reference system at millimeter-level accuracy.

Table IV shows the resulting mean positioning error. Between the original and reconstructed CIRs, the error of the reconstructed signals is 39% higher than the error of the original signals. When using the compressed features, the positioning error more than doubles to 0.97 m. While this difference in performance indicates that the decoder is learning spatially significant information, it is impossible to fully verify this due to the different architectures of the fingerprinting networks. This aspect needs further investigation.

TABLE IV
Fingerprinting Results

Model	Trained/Tested With	Error
FCN-FP-1	Original CIR	0.42 m
FCN-FP-1	Reconstructed CIR	0.59 m
FCN-FP-2	Compressed Features	0.97 m

Since the compression model was trained on a different environment, a slight performance decrease is to be expected. The scenarios are significantly different in their composition and propagation conditions. Combined with the compression rate of 98% this indicates that the fingerprinting approach still provides a robust localization performance. The most important spatial information of the CIR is thus still available in the compressed features. This is a promising initial outcome, as it shows that the compression model works even when switching environments and without any further fine-tuning.

VI. Conclusion

This paper presents and evaluates an approach to compress CIRs with optimized autoencoders. Our large-scale study of possible architectures shows that our fully connected vanilla autoencoders (FCN-AE) outperforms other data-driven variants and the state-of-the-art methods DCT and DWT. Our results serve as a guide for robust and generalizing compression of radio signals. Specifically for FCN-AE we show that two layers are sufficient, with ReLU or ELU activations at the beginning and (hard) sigmoid activations at the end. Compression remains a tradeoff between ratio and error, but a reduction to four components is feasible.

We also provide results for data-driven fingerprinting with a 39% increase in position error paired with a 98% decrease in data transmission. Our results show that a compressed version of the CSI is sufficient to achieve reasonable localization performance when compared to the raw CSI although application-dependent. Future work must determine the role of the decoder model and the full benefits when jointly training autoencoders with task-driven models [21].

Acknowledgment

This work was supported by the Bavarian Ministry for Economic Affairs, Infrastructure, Transport and Technology through the Center for Analytics Data Applications (ADA-Center) within the framework of ”BAYERN DIGITAL II”.

References

[1] P. Cluzeaud, I. Bucaille, A. Sierra, and J. Chóliz, “UWB in Heterogeneous Access Networks: location-aware services,” in *Workshop Positioning, Navi. and Comm.*, (Dresden, Germany), pp. 257–263, 2010.

[2] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, “Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances,” *Sensors*, vol. 16, no. 5, p. 707, 2016.

[3] G. Mendoza Silva, J. Torres-Sospedra, and J. Huerta, “A Meta-Review of Indoor Positioning Systems,” *Sensors J.*, vol. 19, p. 4507, 2019.

[4] T. Feigl, T. Nowak, M. Philippsen, T. Edelhäußer, and C. Mutschler, “Recurrent neural networks on drifting time-of-flight measurements,” in *9th Intl. Conf. Indoor Positioning and Indoor Navigation*, (Nantes, France), pp. 206–212, 2018.

[5] A. Muqaibel, A. Safaai-Jazi, B. Woerner, and S. Riad, "UWB channel impulse response characterization using deconvolution techniques," in *Symp. on Circuits and Systems*, (Tulsa, OK), pp. 587–605, 2002.

[6] P. Meissner and K. Witrisal, "Analysis of position-related information in measured uwb indoor channels," in *European Conf. on Antennas and Propagation*, (Prague, Czech Republic), pp. 6–10, 2012.

[7] T. Jost, W. Wang, U.-C. Fiebig, and F. Perez Fontan, "Detection and Tracking of Mobile Propagation Channel Paths," *IEEE Trans. on Antennas and Propagation*, vol. 60, pp. 4875–4883, 2012.

[8] S. Kram, M. Stahlke, T. Feigl, J. Seitz, and J. Thielecke, "UWB Channel Impulse Responses for Positioning in Complex Environments: A Detailed Feature Analysis," *Sensors J.*, vol. 19, no. 24, 2019.

[9] J. Kulmer, S. Grebien, E. Leitinger, and K. Witrisal, "Delay estimation in presence of dense multipath," *IEEE Wireless Comm. Letters*, 2019.

[10] K. Witrisal and P. Meissner, "Performance bounds for multipath-assisted indoor navigation and tracking (MINT)," in *IEEE Intl. Conf. on Comm.*, (Ottawa, ON, Canada), pp. 4321–4325, 2012.

[11] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Win, "A Belief Propagation Algorithm for Multipath-Based SLAM," *IEEE Trans. Wireless Comm.*, vol. 18, pp. 5613–5629, 2019.

[12] M. Heidari, F. O. Akgul, and K. Pahlavan, "Identification of the Absence of Direct Path in Indoor Localization Systems," in *IEEE Intl. Symp. Personal, Indoor and Mobile Radio Comm.*, (Athens, Greece), 2007.

[13] A. Niitsoo, T. Edelhäußer, and C. Mutschler, "Convolutional neural networks for position estimation in tdoa-based locating systems," in *9th Intl. Conf. Indoor Positioning and Indoor Navi.*, (Nantes, France), 2018.

[14] A. Niitsoo, T. Edelhäußer, E. Eberlein, N. Hadaschik, and C. Mutschler, "A Deep Learning Approach to Position Estimation from Channel Impulse Responses," *Sensors J.*, vol. 19, 2019.

[15] M. Stahlke, S. Kram, C. Mutschler, and T. Mahr, "NLOS Detection using UWB Channel Impulse Responses and Convolutional Neural Networks," in *Proc. Intl. Conf. Localization and GNSS*, (Tampere, Finland), 2020.

[16] K. Ishihara, Y. Asai, R. Kudo, T. Ichikawa, and M. Mizoguchi, "Indoor Experiments of Real-Time MU-MIMO with CSI Feedback Scheme for Wireless LAN Systems," in *IEEE Vehicular Technology Conf.*, (Quebec City, QC, Canada), pp. 1–5, 2012.

[17] P. Wang, J. Wang, C. Zhao, and X. Liu, "A wavelet compression based channel feedback protocol for spatially correlated massive MIMO systems," in *Proc. Intl. Conf. Telecomm.*, (Colmar, France), pp. 303–307, 2015.

[18] E. Leitinger, P. Meissner, C. Rüdisser, G. Dumphart, and K. Witrisal, "Evaluation of Position-Related Information in Multipath Components for Indoor Positioning," *J. Selected Areas in Comm.*, 2015.

[19] Y. Liao, H. Yao, Y. Hua, and C. Li, "CSI Feedback Based on Deep Learning for Massive MIMO Systems," *IEEE Access*, vol. 7, pp. 86810–86820, 2019.

[20] D. Del Testa and M. Rossi, "Lightweight Lossy Compression of Biometric Patterns via Denoising Autoencoders," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2304–2308, 2015.

[21] J. Fontaine, M. Ridolfi, B. V. Herbruggen, A. Shahid, and E. D. Poorter, "Edge Inference for UWB Ranging Error Correction Using Autoencoders," *IEEE Access*, vol. 8, pp. 139143–139155, 2020.

[22] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan, "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications," *Computer and Information Sciences J.*, 2018.

[23] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[24] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Comm. Magazine*, vol. 44, no. 8, pp. 134–143, 2006.

[25] H. G. Musmann, "Genesis of the MP3 audio coding standard," *IEEE Trans. on Consumer Electronics*, vol. 52, no. 3, pp. 1043–1049, 2006.

[26] L. Klus, E. S. Lohan, C. Granell, and J. Nurmi, "Lossy Compression Methods for Performance-Restricted Wearable Devices," *Intl. Conf. on Localization and GNSS*, p. 14, 2020.

[27] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[28] M. Nielsen, E. N. Kamavuako, M. M. Andersen, M.-F. Lucas, and D. Farina, "Optimal wavelets for biomedical signal compression," *Medical and Biological Eng. and Comp.*, vol. 44, no. 7, pp. 561–568, 2006.

[29] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Recognizing Keystrokes Using WiFi Devices," *IEEE J. on Selected Areas in Comm.*, vol. 35, no. 5, pp. 1175–1190, 2017.

[30] X. Shen, G. Yan, J. Yang, and S. Xu, "WiPass: CSI-based Keystroke Recognition for Numerical Keypad of Smartphones," in *Youth Academic Annual Conf. of Chinese Association of Automation*, (Zhanjiang, China), pp. 276–283, 2020.

[31] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep Convolutional AutoEncoder-based Lossy Image Compression," in *2018 Picture Coding Symp. (PCS)*, (San Francisco, CA), pp. 253–257, 2018.

[32] A. Habibian, T. v. Rozendaal, J. M. Tomczak, and T. S. Cohen, "Video Compression With Rate-Distortion Autoencoders," in *Intl. Conf. Computer Vision*, (Seoul, Korea), pp. 7033–7042, 2019.

[33] O. Yildirim, R. S. Tan, and U. R. Acharya, "An efficient compression of ECG signals using deep convolutional autoencoders," *Cognitive Systems Research*, vol. 52, pp. 198–211, 2018.

[34] T. Van Steenkiste, D. Deschrijver, and T. Dhaene, "Interpretable ECG Beat Embedding using Disentangled Variational Auto-Encoders," in *Intl. Symp. Computer-Based Medical Systems*, (Cordoba, Spain), pp. 373–378, 2019.

[35] Y. Jang, G. Kong, M. Jung, S. Choi, and I.-M. Kim, "Deep Autoencoder Based CSI Feedback With Feedback Errors and Feedback Delay in FDD Massive MIMO Systems," *IEEE Wireless Comm. Letters*, vol. 8, no. 3, pp. 833–836, 2019.

[36] H.-C. Tsai, C.-J. Chiu, P.-H. Tseng, and K.-T. Feng, "Refined Autoencoder-Based CSI Hidden Feature Extraction for Indoor Spot Localization," in *IEEE Vehicular Technology Conf.*, (Chicago, IL), 2018.

[37] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep Learning for Massive MIMO CSI Feedback," *IEEE Wireless Comm. Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[38] Z. Cao, W.-T. Shih, J. Guo, C.-K. Wen, and S. Jin, "Lightweight Convolutional Neural Networks for CSI Feedback in Massive MIMO," *IEEE Comm. Letters*, pp. 1–11, 2021.

[39] T. Zhao, F. Li, and P. Tian, "A Deep-Learning Method for Device Activity Detection in mMTC Under Imperfect CSI Based on Variational-Autoencoder," *IEEE Trans. Vehicular Technology*, vol. 69, no. 7, pp. 7981–7986, 2020.

[40] M. Hussien, K. K. Nguyen, and M. Cheriet, "PRVNet: Variational Autoencoders for Massive MIMO CSI Feedback," *arXiv:2011.04178 [cs]*, 2020.

[41] A. Abbasi and H. Liu, "Novel Cascade CNN Algorithm for UWB Signal Denoising, Compressing, and ToA Estimation," in *IEEE Comp. and Comm. Workshop and Conf.*, (Montreal, QC), pp. 721–725, 2021.

[42] C. Nerguizian and V. Nerguizian, "Indoor Fingerprinting Geolocation using Wavelet-Based Features Extracted from the Channel Impulse Response in Conjunction with an Artificial Neural Network," in *IEEE Intl. Symp. Industrial Electronics*, (Vigo, Spain), pp. 2028–2032, 2007.

[43] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proc. Intl. Conf. Learning Representations*, (Banff, AB, Canada), 2014.

[44] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[45] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *Proc. Intl. Conf. Learning Representations*, (Toulon, France), 2017.

[46] X. Qian and D. Klabjan, "The Impact of the Mini-batch Size on the Variance of Gradients in Stochastic Gradient Descent," *arXiv:2004.13146 [cs, math]*, Apr. 2020.

[47] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747 [cs]*, June 2017.

[48] C. Nerguizian, C. Despins, and S. Affes, "Geolocation in mines with an impulse response fingerprinting technique and neural networks," in *IEEE Vehic. Technology Conf.*, (Milan, Italy), pp. 3589–3594, 2004.

[49] T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates," in *Intl. Conf. Machine Learning*, (Atlanta, GA, USA), pp. 343–351, 2013.

[50] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, "A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks," in *Intl. Conf. Comp. Techniques, Elec. and Mech. Sys.*, (Belgaum, India), pp. 92–99, 2018.