Tuning Hyperparameters for DNA-based Discrimination of Wireless Devices

Trevor J. Bihl
Air Force Research Laboratory, USA
Trevor.Bihl.2@us.af.mil

Joseph Schoenbeck
The Perduco Group, USA
joe.schoenbeck@theperducogroup.com

Christopher Rondeau
United States Air Force
christopher.rondeau@afit.edu

Aaron M. Jones
Air Force Research Laboratory, USA
Aaron.jones.41@us.af.mil

Yuki Adams
ARS, USA
yuki.adams.ctr@us.af.mil

Abstract

The Internet of Things (IoT) and Industrial IoT (IIoT) is enabled by Wireless Personal Area Network (WPAN) devices. However, these devices increase vulnerability concerns of the IIoT and resultant Critical Infrastructure (CI) risks. Secure IIoT is enabled by both pre-attack security and post-attack forensic analysis. Radio Frequency (RF) Fingerprinting enables both pre- and post-attack security by providing serial-number level identification of devices through fingerprint characterization of their emissions. For classification and verification, research has shown high performance by employing the neural network-based Generalized Relevance Learning Vector Quantization-Improved (GRLVQI) classifier. However, GRLVQI has numerous hyperparameters and tuning requires AI expertise, thus some researchers have abandoned GRLVQI for notionally simpler, but less accurate, methods. Herein, we develop a fool-proof approach for tuning AI algorithms. For demonstration, Z-Wave, an insecure low-power/cost WPAN technology, and the GRLVQI classifier are considered. Results show significant increases in accuracy (5% for classification, 50% verification) over baseline methods.

1. Introduction

The commercial internet of things (IoT) is enabled by low-cost and low-power Wireless Personal Area Network (WPAN) devices which create mesh networks and allow for widespread interaction and monitoring of smart devices [1]. Due to their abilities, relatively insecure WPAN devices, such as Z-Wave and ZigBee, find their way into Industrial IoT (IIoT) applications, including Critical Infrastructure (CI) uses [2]. Inherent vulnerabilities exist in WPAN technologies, see [3], which is compounded due to device-to-internet pathways, sensitivity of the CI applications, and that one compromised device can threaten the security of the entire network [4].

Robust security is of interest for both pre-attack defense and post-attack forensics by improving the ability to determine WPAN device identities. A general understanding of ISO layers (NWK: Network, MAC: Media Access Control, and PHY: Physical) and their relationship to security can be considered as [5]:

1. "Something you know" (NWK – encryption keys)
2. "Something you have" (MAC – MAC address)
3. "Something you are" (PHY – RF Fingerprints)

However, predominantly, WPAN security considers bit-level identities (MAC and NWK); thus ignoring intrinsic properties found at the PHY layer.

Whether addressing pre-attack defense or post-attack forensic analysis, the preponderance of threat detection and protection work in process control systems occurs above the PHY layer. Radio Frequency (RF) fingerprinting involves computing features, "fingerprints," for predefined signal regions, such as preambles, by dividing the signal into bins and computing statistical features for each bin [6]. For this, the standard three step biometric process (library building, classifier model development, and verification) [7], is followed. Key to this are accurate machine learning (ML) methods.

The Generalized Relevance Learning Vector Quantization-Improved (GRLVQI) classifier, a nonlinear Artificial Neural Network (ANN) algorithm, has been shown to provide good discrimination ability for RF Fingerprinting [5]. However, GRLVQI, as with other ANN algorithms, has multiple hyperparameters, which require luck or expertise to tune effectively. In general, there are "no hard-and-fast rules" in determining hyperparameter and their selection is part of the "art of [algorithm] design" [8]. Prior works in related areas such as feature selection, have even cited the complexity of the task of hyperparameter selection and tuning as a reason to utilize more simple ML algorithms [9]. In some cases - particularly in post-attack forensics where the amount of data may be very large – this could limit the utility of more powerful computing paradigms, such as ANNs. This is further exacerbated by the difficulty of data collection - and even attribution of results - from Operational Technology (OT) systems in Supervisory Control and

URI: https://hdl.handle.net/10125/71458
978-0-9981331-4-0

Data Acquisition (SCADA) and CI environments where specific system expertise is most likely required [10]. To that end, quality ML models are often hand-crafted and require significant expertise (i.e., luck and talent) to appropriately train and deploy. Care is needed in the specification of ML models too, since an overly conservative learning rate results in sub-optimal performance preventing convergence. However, an overly liberal learning rate could result in highly oscillatory training behavior, again, with sub-optimal performance. Thus, to serve the pre-attack or post-attack needs of an IIoT system, a global goal of any proposed technique should be to maximize the efficiency of the algorithm, harness the full power of the best available algorithms for the task, and minimize the expert knowledge – both in system and algorithm.

Prior work, c.f. [5] [6], examined full factorial design of experiments (DoE) and hill-climbing approaches. However, both of these approaches had significant limitations. DoE methods are comptuationally costly and only explore limited regions of the operating space and while hill-climbing methods quickly become trapped by local optima. An additional limitation is neither can handle both discrete and continuous variables. Recently, Bayesian Optimization (BO) has been shown to be superior to other hyperparameter determination methods in both efficiency and model accuracy. BO exploits the randomness inherent in stochastic processes, such as ANNs, and finds viable operating points.

The contributions of our paper are as follows. This work compares four hyperparameter optimization methods for WPAN security using the GRLVQI algorithm as the representative classifier. The four methods are BO, leveraging the process of [11], Stochastic Approximation (SA) [6], and DoE [5]. An extension of CRISP-DM is used to create a repeatable process for this purpose using experimentally collected Z-Wave RF Fingerprints. We apply the four hyperparameter optimization methods to GRLVQI and aim to make the experiments as similar as possible. Evaluations consider both classification (1 vs N) and verification (1 vs 1 claimed identity) with results showing the BO-optimized GRLVQI outperforms past work by 50% in true verification rate accuracy. We further illustrate how BO offers better accuracy, easier operations than other methods, and greater understanding algorithmic hyperparameter space.

2. Background

With expansions of the IoT, the cyber attack surface is increasing due to the multitude of sub-internet pathways. Some examples are common WPAN technologies such as WiFi, Z-Wave, ZigBee and Bluetooth devices. This includes expansions of the IIoT into SCADA [12] systems and CI. Problematically, WPAN devices serve as the backbone for IoT and IIoT connectivity and these often have notable security deficiencies.

Related to the WPAN devices, the industrial systems WPAN technologies thrust into the IIoT fall under the banner of OT, as noted in Section 1. Many, if not nearly all, OT systems in operation were desiged to operate, sense, or monitor an industiral process safely and reliably [13]. As the IoT and OT systems converge in the IIoT, the introduction of OT to these wireless networks has outpaced the inclusion of adequate security measures, or been unable to simply adopt a known cybersecurity practice [13]. Furthermore, OT systems often require system-by-system expertise [10].

The expert knowledge paradigm within OT is an obvious drawback for any system defense framework or forensic analysis technique, it opens an opportunity for exploiting knowledge of the better known WPAN technologies. One such WPAN technology used in IIoT applications is Z-Wave.

2.1. Z-Wave Devices

Z-Wave is a WPAN technology that offers both small and low-cost hardware devices that support many network topologies [14]. Practically, Z-Wave is similar to ZigBee and other technologies, but is simpler to work with [14], differences also exist in security, operating frequency, data rate, and latency. Primarily, the proprietary nature of the Z-Wave standard and lack of initial encyption result in Z-Wave being less secure than competing WPAN technologies [15]. Basic knowledge of Z-Wave suggests it has a similar ISO architecture to ZigBee due to its adherence to the ITU-T G.9959 protocol at the PHY and MAC layers [16]. However, as illustrated in Figure 1, details above the MAC layer are unknown. Therefore, to understand Z-Wave, digital forensics, c.f. [17], are necessary and an emerging area of interest [3].

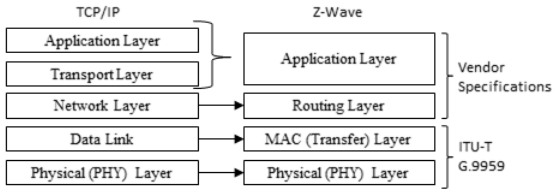

Figure 1. Z-Wave device protocol characteristics, from [6].

The PHY packet structure of Z-Wave is conceptualized in Figure 2. This illustrates two critical components of the Z-Wave protocol: the predefined preamble and the Start of Frame (SoF) [18]. Z-Wave

is also known to include a 32-bit payload-based home identification and 8-bit source identification [15].

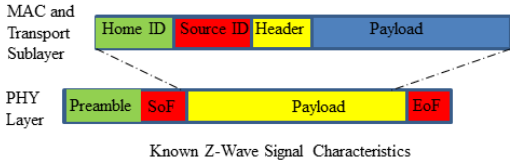

Figure 2. Z-Wave signal characteristics, from [6].

2.2 RF-DNA Fingerprinting

RF Fingerprinting was implemented per the RF-DNA (Distinct Native Attributes) fingerprinting process of [19]. RF-DNA is a systematic process of RF Fingerprinting that involves selecting an ROI to extract, then digital filtering, along with computating the instantaneous amplitude, frequency and phase, fingerprint generation, and finally classifier model development and verification testing [19]. RF-DNA provides biometric-like security of communication devices with discrimination abilities at the serial-number level.

2.2.1. Z-Wave Signal Collection and Pre-Classification Signal Processing. Of interest in RF Fingerprinting of Z-Wave devices is exploiting the knowledge that the Preamble and SoF have a predefined and known order of 1s and 0s which should be the same for all Z-Wave devices. Thus, we aim to discriminate between individual Z-Wave devices based on minute variations in the preamble signals.

As described in [5] [6], to create a Z-Wave database, three devices (N_D = 3), were considered with each device being an Aeon Labs' Aeotec Z-Stick S2 transmitters. For each device, a total of 230 preamble responses were collected [5]. The preamble response (the first 8.3 ms of Z-Wave bursts) was the region of interest (ROI) for studying this data and thus devices were turned on/off to collect preamble data without necessarily completing package delivery [5].

As discussed in [5] [6], for data collection each device was placed 10 cm in line of sight from a vertically-oriented log periodic antenna (LP0410, Ettus Research, Santa Clara, CA). The antenna was connected via a Gigabit Ethernet cable directly to a software defined radio device RF input (USRP-2921, National Instruments) [5] [6].

Signals were collected with a sample frequency of f_s = 2 Msps along with the bursts detected via an amplitude-based leading edge detector with a -6 dB threshold [5] [6]. The collected bursts had a native Signal-to-Noise Ratio (*SNR*) at SNR_C = 24.0 dB amd were liked filtered [6]. To replicate more real world (degraded and distant) conditions, and consistent with [20], independent like-filtered Additive White Gaussian Noise (AWGN) was applied to achieve operating conditions of $SNR \in [0\ 24.0]$ dB in 2 dB increments [6].

Due to the size of the data, i.e. only 3 devices considered, and the manual workload required to collect additional data, all devices were considered as serving in "authorized" roles. Thus, impersonation attacks by "rogue" devices are not considered

2.2.2. Fingerprint Generation. Consistent with [20], RF-DNA fingerprint generation begins by computating the instantaneous time domain responses of amplitude (a), phase (ϕ), and frequency (f) for the signal. These responses are then divided into N_R contiguous and equal length bins and then N_s = 3 features of variance (σ^2), skewness (γ), and kurtosis (κ) are computed [19], [20]. As conceptualized in Figure 3, N_s features are computed for each bin and across the entire response for a total of N_R + 1 features per amplitude, phase, and frequency response at a given *SNR* [19], [20]. From this, one considers RF regional fingerprint vectors as

$$\boldsymbol{F}_{Ri} = [\sigma^2_{Ri}, \gamma_{Ri}, \kappa_{Ri}]_{1\times 3}\ , \quad (1)$$

where $i = 1,2,\ldots,N_R+1$, for the N_S = 3 RF fingerprint features (statistics) [19], [20].

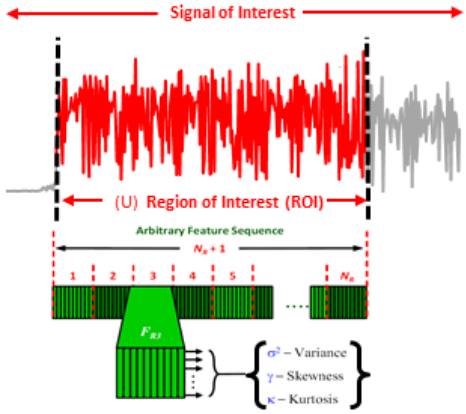

Figure 3. RF fingerprinting concept, from [20].

From RF regional fingerprint vectors, a fingerprint vector for each of the responses is formed from (1) as,

$$\boldsymbol{F}^C = \left[\boldsymbol{F}_{R_1}\ \boldsymbol{F}_{R_2}\cdots\boldsymbol{F}_{R_{(N_R+1)}}\right]_{1\times N_s(N_R+1)}\ , \quad (2)$$

which are concatenated to form the fingerprint vector:

$$\boldsymbol{F} = \left[\boldsymbol{F}^{\boldsymbol{a}} : \boldsymbol{F}^{\boldsymbol{\phi}} : \boldsymbol{F}^{\boldsymbol{f}}\right]_{1\times N_s(N_R+1)\times N_C}\ . \quad (3)$$

Consistent with [5] [6], N_R = 20 subregions spanning the ROI were considered for Z-Wave devices. Given the 230 preambles collected per Z-Wave device and the N_R, a total of N_F = 189 features (a, ϕ, and f responses with σ^2, γ, and κ features) were computed with equal splitting of the data between training and testing in an interleaved manner with N_{TRN} = 115 Training (TNG) and N_{TST} = 115 Testing (TST)

preamble observations per device. Thus, with $N_C = 3$ devices, our dataset has a total of $N_{TRN} = 345$ and $N_{TST} = 345$ observations, each with $N_F = 189$ fingerprint features. To avoid the possibility of overfitting, TNG and TST data were sequestered.

2.3. Classifier Models

Various classifiers have been applied to RF and RF-DNA fingerprints, including Multiple Discriminant Analysis (MDA) [19], GRLVQI [5], and random forests [21]. Of interest herein is GRLVQI which has 1) a well known trackrecord for valid RF Fingerprinting classification [5], but also 2) a variety of hyperparameters to select. MDA is of further interest to provide a performance baseline. Furthermore, both MDA and GRLVQI have their own performance advantages in RF fingerprinting problems [5] [6].

2.4.1. GRLVQI Classifier Model. GRLVQI belongs to the neural network family of algorithms known as self-organizing ANNs [22]. GRLVQI is an extension, see [8], of the Learning Vector Quantization (LVQ) approach of Kohonen [23]. LVQ methods employ *nearest neighbor* approaches through the *nearest node*, or *prototype vector* (PV) in LVQ terminology, whereby each PV is iteratively moved to characterize the data through a lower dimensionality structure that captures the data's characteristics [24]. Practically, LVQ algorithms train PVs to a given class label by moving correctly classified PVs closer to a given class and moving incorrectly classified PVs away.

LVQ has seen many embellishments which create new algorithms [25], GRLVQI is one such algorithm and the additional letters in the acronym signify each embellishment: *G (generalized)* for the inclusion of a sigmoidal cost function [26], *R (relevance)* for the incorporation of an extra loop for relevance learning [27], and *I (improved)* for improvements in PV update logic and operation [8]. The improvements of GRLVQI over GRLVQ [27] include the conscience learning of DeSieno [28], improved PV update logic, and a frequency based maximum input update strategy [8].
When compared to the simple conceptualization of LVQ, GRLVQI is much more complicated. Both employ a gradient descent learning rate (ϵ) to determine how fast the PVs move [23], and the number of prototype vectors (N_{PV}) per class determine network size. GRLVQI also has a relevance learning rate (ξ) to determines how quickly variables are penalized for being possibly insignificant [27]. Additionally, GRLVQI employs conscience rates (γ and β) determine how frequently individual PVs should be moved [8]. Thus, GRLVQI has 5 hyperparameters to consider. Additionally, GRLVQI is stochastic in nature, as are ANNs in general, and results may vary based on random selection of training data.

2.4.2. Multiple Discriminant Analysis (MDA). MDA is a linear approach to classification which considers an eigenvector-based projection of the data relative to a ratio of between-group to within-group sum-of-squares, known as the Fisher criterion [19]. For RF-DNA fingerprints, MDA considers input fingerprint matrix $\boldsymbol{F}$ and N_C classes. Thus, MDA is largely intuitive in how it operates, is not stochastic, and it is also computationally inexpensive. However, MDA can encounter difficulties with highly nonlinear data or when the number of features approaches, or exceeds, the number of observations. For Z-Wave data, MDA generally underperforms GRLVQI in classification, but outperforms it in verification [5].

2.5. Quantifying Algorithm Performance

To evaluate WPAN device security, two general considerations exist: classification and verification. Classification considers "one vs. many," and is evaluated using confusion matrices for classifier models trained at each *SNR* operating point [19]. Verification is considered as a "one vs one" claimed identity scenario for a classifier model at a specific *SNR* with the signal compared using the classifier model and the associated probability mass function [5].

2.5.1. Classification Metrics. Classification performance is evaluated using two metrics, gain and Relative Accuracy Percentage (RAP), from examining a plot of average percent correct classification (*%C*) versus *SNR* [5].

Gain is defined as the reduction in required *SNR*, expressed in dB, for two methods to achieve the same *%C*, i.e. an arbitrary performance benchmark of *%C* = 90% [5] [19]. Per [5], gain values, G_{SNR}, is interpretted:
1) $G_{SNR} < 0.0$ (negative), a method achieves the same *%C* at a higher *SNR*, i.e. it underperforms.
2) $G_{SNR} = 0.0$, a given method achieves the same *%C* as the baseline at the same *SNR*
3) $G_{SNR} > 0.0$ (positive), a method achieves the same *%C* at a lower SNR, i.e. it outperforms.

However, *gain* only considers one part of the *%C* vs. *SNR* curve and it might not have a value if a method never reaches 90%. Thus, Relative Accuracy Percentage (RAP) and Area Under Classification Curve (*AUCC*) were introduced in [5]. RAP values are the computed by finding the AUCC for perfect accuracy across the x-axis and then dividing $AUCC_{method}$ by $AUCC_{perfect}$. RAP is compared with higher RAP indicating better overall accuracy.

2.5.2. Verification Metrics. Verification, as described in [19], involves: 1) an unknown device claiming bit level credentials (e.g., MAC address) which match a specific authorized device, 2) extracting RF fingerprint features from the unknown device, and 3) comparing new RF fingerprints against the model. Verification performance is evaluated using Receiver Operating Characteristic (ROC) curves at a specified *SNR* for True Verification Rate (TVR) versus False Reject Rate (FRR); in experimentation this is typically at the lowest *SNR* a model achieves *%C* = 90% [5]. Two metrics can be computed [5]: firstly, the percentage authorized (*%Aut*) at TVR ≥ 90% at FVR ≤ 10%. However, *%Aut* is coarse and dichotomous, e.g. N_D = 3 devices *%Aut* ∈ [0, 33, 66, 100], and thus distinguishing between relative performance differences is impossible, thus we also consider the mean area of the ROC curves (AUC_M) [5].

3. Hyperparameter Optimization

Hyperparameter determination is an emerging discipline in AI and includes a multitude of methods. A general taxonomy of these approaches is presented in [11]. These can largely be separated into model-free and model-based approaches [29].

Model-free approaches can be 1) scientific, e.g. grid searches, or 2) haphazard, e.g. a coder experientially finding settings that "just work," or 3) random searches which use random seeds (notably a competitive method). Model-based approaches employ a wrapper on an outer loop around the algorithm of interest and determine settings to explore in a concerted search strategy. From the families of model-based approaches listed in [11], we consider:

- **Stochastic Approximation** [6], which is a hill climbing approach with hyperparameters individually and sequentially changed. Previously applied to GRLVQI in [6].
- **Bayesian optimization (BO)** [30] whereby the objective function is treated as a random function and randomly determined hyperparameters taken from the appropriate distribution around the results are found. BO tends to find reasonably good choices of hyperparameters, but this has not been rigorously studied for cyber applications yet

The concern with all such hyperparameter optimization methods is finding *locally* optimal solutions, which are potentially significantly different from the *globally* optimal solutions. Unless one explores all possible setting combinations, which is often impossible, it is never possible to be certain that one has arrived at the globablly optimal solution. Thus, the different algorithm families aim to address this through different strategies: replications in DoE, replications and embracing randomness in model-based approaches, etc.

3.1. Grid Search Approaches

A grid search involves creating a set of design point combinations to explore [31]. Examples include ad hoc approaches and factorial designs [31]. Factorial experiments consider all combinations to understand significance of factors, interaction of factors, and to find viable operating points. The state of the art in DoE includes space-saving designs which explore a logically chosen subset of all combinations of settings.

Grid searches can involve multiple steps as well. As described in [5], these steps can include:

1. Initial design execution
2. Optimal solution
    a. Spreadsheet search
    b. Response surface methods (RSMs)

For an algorithm such as GRLVQI, there could be 243 (3^5) models to create and evaluate in order to explore the combinations from 3 settings for each of the 5 factors [5]. The spreadsheet search then is merely finding the best result from these runs [5]. The RSM further explores the space by considering an analysis of variance of the factors and their interactions and then by fitting a second order model to the data [5]. From this, new algorithm settings are computed to explore and, hopefully, a better result is found [5].

3.2. Stochastic Approximation

A general stochastic approximation method is the Kiefer and Wolfowitz approach of sequential design [6]. Here, we will let $h_{i,j}$ be the value of the i^{th} of N continuous valued hyperparameter of the function of interest at iteration j of an optimization procedure. From this, $\boldsymbol{h}_j$ is a vector of these hyperparameters. Let $f(\boldsymbol{h}_j)$ be the performance measure of interest of our function of interest. Finally, let $\{a_n\}$ and $\{c_i\}$ be sequences

$$\begin{gathered}\sum_{i=1}^{\infty} a_i = \infty\,, \\ \sum_{i=1}^{\infty} a_n c_n < \infty\,, \\ \sum_{i=1}^{\infty} a_n^2 c_n^{-2} < \infty\ .^{1}\end{gathered} \tag{5}$$

Using (5), let $\boldsymbol{c}_j^i = \left(\mathbf{0}^{i-1}, c_j, \mathbf{0}^{N-i}\right)$ where $\mathbf{0}^n$ is a vector of zeroes of size n. The algorithm works as

[1] Suggested sequences are $\{a_n\} = \frac{1}{n}$ and $c_n = n^{-\frac{1}{3}}$

follows: with $\boldsymbol{\tau}$ being a termination criteria and $\boldsymbol{f}(\boldsymbol{h_i})$ representing the objective function value. Also, let $|\boldsymbol{h_i} - \boldsymbol{h_{i-1}}|$ be the $\mathbf{L^1}$ norm of $\boldsymbol{h_i}$ and $\boldsymbol{h_{i-1}}$. The algorithm then iterates over

$$\begin{bmatrix} h_{1,j+1} \\ \vdots \\ h_{i,j+1} \\ \vdots \\ h_{n,j+1} \end{bmatrix} = \frac{a_i}{2c_i} \begin{bmatrix} f(\boldsymbol{h}_j + \boldsymbol{c}_j^1) - f(\boldsymbol{h}_j - \boldsymbol{c}_j^1) \\ \vdots \\ f(\boldsymbol{h}_j + \boldsymbol{c}_j^i) - f(\boldsymbol{h}_j - \boldsymbol{c}_j^i) \\ \vdots \\ f(\boldsymbol{h}_j + \boldsymbol{c}_j^N) - f(\boldsymbol{h}_j - \boldsymbol{c}_j^N) \end{bmatrix} \quad (6)$$

with one hyperparameter at a time changed. The algorithm terminates when the norm of the differences between $f(\boldsymbol{h})$ of two consective iterations is small. In operation, the process changes one individual hyperparameter's value at a time and looks at objective function results when an upper, $f(\boldsymbol{h}_j + \boldsymbol{c}_j^1)$, and lower, $f(\boldsymbol{h}_j - \boldsymbol{c}_j^1)$, value of the hyperparameter is used.

3.3. Bayesian Optimization (BO)

BO similarly considers $h_{i,j}$ and $\boldsymbol{h}_j$; but additionally, $h_{i,j}$ is in the bounded set $\mathcal{H}_i$, which can be continuous or integer valued. BO notes that $f(\boldsymbol{h}_j)$ is stochastic in nature due to the randomness in the results as a function of the random selection of training data [11]. For BO, let $y_j = f(\boldsymbol{h}_j)$ and let $\{y_j, h_j\}_{j=1}^n$ be a sequence of y_j and h_j pairs. Based on this sequence, a Gaussian process can be fit to $f(\cdot)$, denoted by $GP\left(\{y_j, h_j\}_{j=1}^n\right)$. Finally, $a\left(\boldsymbol{h} \middle| GP\left(\{y_j, h_j\}_{j=1}^n\right)\right)$, an acquisition function, is maximized to find a new set of candiate hyperpameters. While $a(\cdot)$ can be chosen by the BO designer, common choices are expected improvement, probability of improvement, and lower confidence bounds [11]; expected improvement was used herein.

Sequentially, BO follows the following steps [11]:

1. Obtain n_0 initial evaluations of $f(\cdot)$ at randomly selected values of hyperparameters within the specified hyperparameter bounds. Set $k = 0$.
2. Fit a Gaussian Process onto $\{y_j, h_j\}_{j=1}^{n_k}$, denoted as $GP\left(\{y_j, h_j\}_{j=1}^{n_k}\right)$.
3. Set $\boldsymbol{h}_{j+1} = \underset{\boldsymbol{h}}{\operatorname{argmax}}\, a\left(\boldsymbol{h} \middle| GP\left(\{y_j, h_j\}_{j=1}^{n_k}\right)\right)$
4. Evaluate $y_{j+1} = f(\boldsymbol{h}_{j+1})$, set $n_k = j + 1$ and $k = k + 1$. If termination criteria $\boldsymbol{\tau}$ is not met, go-to step 2.

3.4. CRISP-DM+ Approach for Hyperparameter Optimization

Approaches such as CRISP-DM, provide general end-to-end processes to develop ML solutions [11]. Work in [11] extended CRISP-DM by expanding the Data, Modeling, and Evaluation layers to include 1) selecting a dataset, 2) selecting an AI algorithm, and then 3) automatically determining hyperparameter settings without expert algorithmic knowledge. Herein, we consider the CRISP-DM+ items of [11]:

A1. Data Wrangling for the Z-Wave RF-DNA problem, this was presented in Sections 2.2 and 2.3.

A2. Select ML Architecture this was presented in Sections 2.4 and 2.5 for both the GRLVQI algorithms and the performance metrics of interest.

B1. Train ML Model Using Default Weights generally involves taking the algorithm from A2, and finding a starting point from either default settings or example settings from help documentation [11]. Herein, this involves baselining with known settings.

B2. Optimize Hyperparameters involves finding reasonable settings via hyperparameter optimization methods. Of concern is determining initial ranges for the weights for the hyperparameter optimization methods. For BO, this could include the extreme limits of the design space, but not for SA and grid searches.

C. Test & Compare Optimized Model. This will consider the classification and verification evaluation methods previously described in Section 2.5.

4. Experimental Results

To provide a general comparison of methods, this work considers four different methods of hyperparameter optimization: 1) a grid-search using a full factorial design and a spreadsheet search for the best result (GRLVQI-SS), 2) a response surface method extended upon the factorial design (GRLVQI-RSM), 3) stochastic approximation (GRLVQI-SA), and 4) BO (GRLVQI-BO). This is compared against both 5) the baseline experientially determined settings of [5] (GRLVQI-base), and 6) MDA.

4.1. Hyperparameter Design Region Considerations

Although hyperparameter optimization removes the problems of finding initial algorithm settings, the new problem is determining bounds for the search

region for each hyperparameter. As discussed in [5], limited prior work on GRLVQI hyperparameters for any purpose, let alone RF Fingerprinting, exists. Some general guidelines on settings do exist, and these include 1) values should be non-negative (negative learning rates would cause PVs to deviate from learning goals), and 2) a general recommendation that $0 \leq \xi(t) \leq \epsilon(t) \leq 1$ [5]. Similar guidance does not exist for the conscience parameters, beyond non-negativity. Additionally, the only general guidance on PVs is that too few will not capture the data well and too many will overfit. Capturing these guidelines, using limits outside previously explored conscience rates, we have the general search region displayed in Table 1.

Table 1. General Hyperparameter Search Region for GRLVQI

Param.	Meaning	Initial Search Interval
ϵ	Learning Rate	[0, 1]
ξ	Relevance Learning Rate	[0, 1]
γ	Conscience Rate 1	[0, 10]
β	Conscience Rate 2	[0, 2]
N_{PV}	Number of Prototype Vectors (PVs) per class	[2, 28]

As seen in [5], traditional DoE need meaningful bounds to explore since the combination of settings from the extreme points of the interval are explored. Thus, Table 1 would be an impossibly wide interval with mostly unusable results (from learning rates of 0). Thus, the work of [5] explored values near the experimentally determined $\boldsymbol{h_0} = (0.025, 0.005, 2.5, 3.5, 10)$ per [5]. For this, 243 (3^5) points were explored (high, centered, low).

BO and SA operate different than the DoE. Both can explore the space but do so in a different manner. BO will begin to model the response as a random process and collect seemingly random observations; SA will implement a hill-climbing approach and look for individual improvements to each hyperparameter. BO can explore the entire space of Table 1 and was allowed to do so. SA needs a good initial operating point to improve upon; thus, as presented in [6], SA approached the problem by improving up on the experimentally determined $\boldsymbol{h_0} = (0.025, 0.005, 2.5, 3.5)$ [6]. For comparison across methods, the same computational "budget" of 3^5 design points was given to BO and SA as well. In SA much of this is spent by exploring the 4 continuous hyperparamters $(\epsilon, \xi, \gamma, \beta)$ with changes of ±c as seen in (6). Thus, SA could only perform 31 full iterations (248 points) while BO could explore 243 unique points. In all cases, maximizing RAP was the objective and all algorithms were optimized for TNG set performance.

Table 2. Experimental design region for GRLVQI from [5]

Param.	Meaning	Search Interval
ϵ	Learning Rate	[0.0025, 0.025, 0.25]
ξ	Relevance Learning Rate	[0.0005, 0.005, 0.05]
γ	Conscience Rate 1	[0.5, 2.0, 4.5]
β	Conscience Rate 2	[0.15, 0.35, 0.55]
N_{PV}	Number of Prototype Vectors (PVs) per class	[7, 10, 13]

4.2. Results

Figure 4 presents the classification performance from the optimal from each method using the TST set of *%C* versus *SNR* for 1) GRLVQI-BO, 2) GRLVQI-SA), 3) GRLVQI-RSM, 4) GRLVQI-SS, 5) GRLVQI-base, and 6) MDA. Notably, GRLVQI outperforms MDA, and each optimization method providing further improvements with BO being the best.

Table 3 condenses the results and presents verification performance. In Table 3, RAP values were

Table 3. Hyperparameters optimization comparative results, performance results in bold indicate best or within 5% of the best by column

Method	N_{RUNS}	Factors Levels					Performance Results					
							Classification				Verification at *SNR* = 20dB	
		A	B	C	D	E	G_{SNR} (dB) at %C = 90%		RAP (%)		TVR (%)	AUC_M
							TNG	TST	TNG	TST		
GRLVQI-BO*	243	0.868	0.0014	6.881	0.392	3	**+7.02**	**+5.89**	**69.21**	**68.57**	**100**	**0.982**
GRLVQI-SA	248†	0.078	0.016	2.527	0.319	7	+5.16	+5.05	65.33	64.79	66	**0.965**
GRLVQI-SS	243	0.25	0.05	2.0	0.35	7	+5.30	**+5.77**	67.39	**65.80**	66	**0.979**
GRLVQI-RSM	249	0.150	0.05	4.5	0.15	7	+5.23	+5.26	66.57	**65.33**	66	**0.967**
GRLVQI-base	N/A	0.025	0.005	2.5	0.35	10	+3.72	+3.32	62.63	61.26	33	**0.936**
MDA	N/A	N/A					+1.68	0.00	**68.27**	55.5	**100**	**0.971**

*Proposed herein.

†For 31 iteration; it should be noted that [6] also performed 10 replications per iteration

computed relative to perfect results and thus RAP is a percentages of the area under a method's *%C* vs *SNR* plot. Here we see that GRLVQI outperforms MDA and the baseline GRLVQI significantly with progressively better classification performance as one moves up the table. BO notably provides considerably better performance across all classification metrics.

Additionally, Table 3 includes verification accuracy performance of all algorithms. Non-intuitively, GRLVQI has generally underperformed MDA at verification. Prior hyperparameter optimization attempts, see [5] [6], improved both verification and classification performance, but could not achieve *%C* = 100% authorized. However, the best design point from BO was able to achieve *%C* = 100% while outperforming all other methods for classification. Thus, the BO optimized GRLVQI offers considerably improved performance over baseline GRLVQI and MDA which was achieved with a reasonable computation budget.

4.2. Results, Digging Deeper

As seen in Table 3 and noted in [6], very different combinations of settings can yield acceptable results. Thus, it is expected that multiple local maxima exist. To further investigate this, we can explore the surface of design points and results. Figure 5 presents this surface for RAP versus the learning rate and relevance learning rate. Blue dots are the explored points and the surface is interpolated between points; the best value obtained by a small red X in the lower right. Figure 5 shows that the highest RAP values are located in different areas of the parameter space with the surface itself is surprisingly variable.

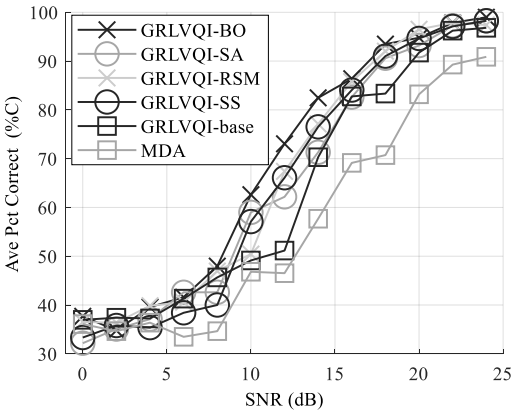

Figure 4. Classification performance for the TST set on evaluated algorithms

Further exploration of the hyperprameter space is seen in Figure 6 for RAP versus conscience rates, and Figure 7 for RAP versus learning rate and N_{PV}. These surfaces exhibit a similarly high variablilty and illustrate the difficulty in finding acceptable operational settings.

Figure 8 overlays both the DoE points (red x's) of Table 2 and the SA design points (black o's) onto a subset of Figure 5. The path of SA shows the sequential approach of this method and SA notably explored only a small region of the space. Conversely, the DoE approach is seen to explore more of the space, but demonstrates an inefficiency in that many runs appear wasted due to the design being full factorial in nature. Thus, these data suggest that a space saving design would be more efficient. However, as Figure 8 shows that both SA and DoE explored only a small region of Figure 5, BO has further advantages in exploring more of the hyperparameter space.

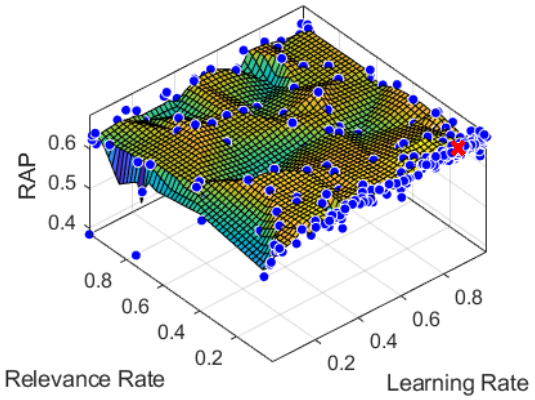

Figure 5. RAP results versus learning rate and relevance rate. GRLVQI-BO is red x (right).

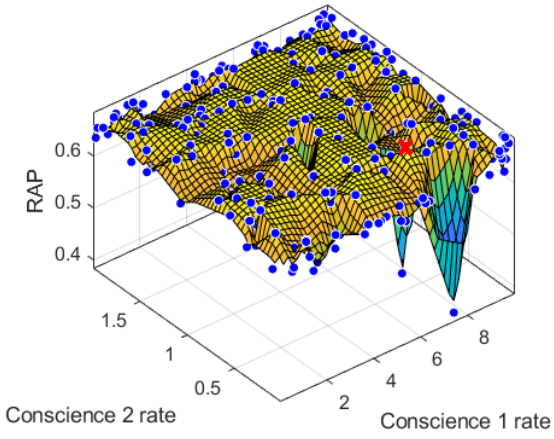

Figure 6. RAP results versus Conscience rate 1 and Conscience rate 2. GRLVQI-BO is red x (middle right).

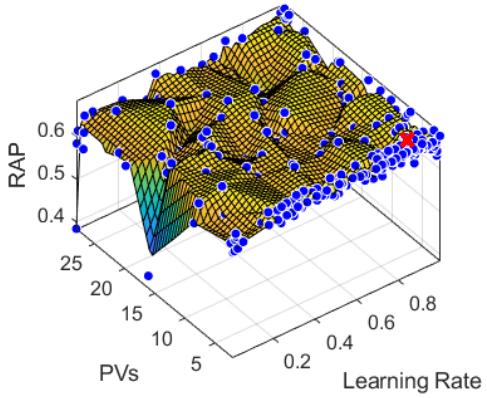

Figure 7. RAP results versus learning rate and PVs. GRLVQI-BO is red x (right).

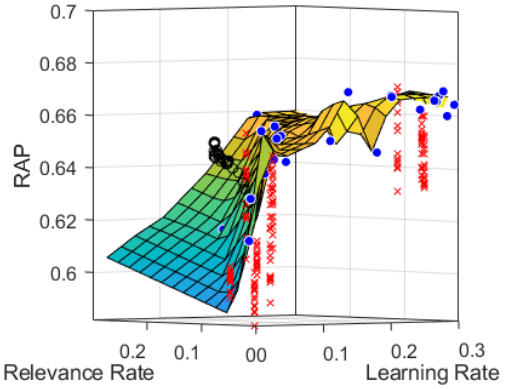

Figure 8. Overlay of SA (black O's) and DoE (red X's) approaches to hyperparameter optimization onto a small region, the lower middle, of Figure 5.

5. Conclusions

The authors presented the systematic application of hyperparameter optimization for WPAN device identification. For an application baseline, experimentally collected Radio Frequency (RF) Fingerprints for Z-Wave devices were considered. The GRLVQI neural network algorithm, which has 5 tunable hyperparameters, was considered due to its prior successes in RF Fingerprint identification. However, no previously explored methods, including GRLVQI, achieve suitable performance consistently in both classification (one vs many) and verification (one vs one claimed identity). To improve the operational security of GRLVQI for RF Fingerprinting applications, and other algorithms in general, we explored the application of four hyperparameter optimization methods to finding good settings.

This work illustrated the necessity in determining appropriate GRLVQI algorithm settings and provided a further understanding of the hyperparameter and response relationship. Primary contributions include improvements to communication device discrimination using RF Fingerprints by: 1) applying a CRISP-DM+ approach to hyperparameter optimization for RF Fingerprinting, 2) demonstration of this approach for GRLVQI optimization for Z-Wave device discrimination, 3) improvements in the experimental approach of RF Fingerprinting classifier development, 4) an understanding of the hyperparameter space for complex cyber problems and algorithms, and 5) a comparison of 4 hyperparameter optimization methods. In total we compared Bayesian Optimization (BO), Stochastic Approximation (SA), Design of Experiments (DoE) with both a Spreadsheet Search (SS) and Response Surface Methodology (RSM). The systematic application of Bayesian Optimization (BO) was able to find GRLVQI algorithm settings that exceeded all prior bests at both classification and verification with 100% verification accuracy achieved. The results further showed limitations in SA and DoE-based approaches which explored considerably more limited regions of the hyperparameter space.

The theme of future work, in general, considers deeper understanding of the hyperparameter tradeoff space, including evaluating the robustness of these GRLVQI hyperparameters for other WPAN devices. Future work could also include combining methodologies, like refining initial BO solutions with SA, and appropriately handling variables that vary logarithmically, linearly, and as integers. Additionally, broader comparisons with other optimization methods additionally needs to be considered.

6. Acknowledgement

The views expressed herein are those of the authors and do represent any position of their employers. This was approved for public release under case: 88ABW-2020-2230.

7. References

[1] A. Wong, "Case Study: Simulated deployment of a mesh network in Honolulu," *Hawaii International Conference on System Sciences (HICSS)*, pp. 1-9, 2010.

[2] V. Güngör, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati and G. Hancke, "Smart grid technologies: communication technologies and standards," *IEEE Trans. on Industrial Informatics*, vol. 7, no. 4, pp. 529-539, 2011.

[3] C. Badenhop, B. Ramsey, B. Mullins and L. Mailloux, "Extraction and analysis of non-volatile memory of the ZW0301 module, a Z-Wave transceiver," *Digital Investigation*, vol. 17, pp. 14-27, 2016.

[4] H. Yang, H. Luo, F. Ye, S. Lu and L. Zhang, "Security in mobile ad hoc networks: challenges and solutions," *IEEE wireless communications*, vol. 11, no. 1, pp. 38-47, 2004.

[5] T. Bihl, M. Temple and K. Bauer, "An Optimization Framework for Generalized Relevance Learning Vector Quantization with Application to Z-Wave Device Fingerprinting," *Hawaii International Conference on System Sciences (HICSS)*, pp. 2379-2387, 2017.

[6] T. Bihl and D. Steeneck, "Multivariate Stochastic Approximation to Tune Neural Network Hyperparameters for Criticial Infrastructure Communication Device Identification," *Hawaii International Conference on System Sciences (HICSS)*, pp. 2225-2234, 2018.

[7] S. Prabhakar, S. Pankanti and A. K. Jain, "Biometric recognition: Security and privacy concerns.," *IEEE security & privacy*, vol. 99, no. 2, pp. 33-42, 2003.

[8] M. J. Mendenhall, *A Neural Relevance Model for Feature Extraction from Hyperspectral Images, and its Application in the Wavelet Domain*, PhD Dissertation: Rice University, 2006.

[9] C. M. Rondeau, M. A. Temple and C. S. Kabban, "TD-DNA Feature Selection for Discriminating WirelessHART IIoT Devices," *Hawaii International Conference on System Sciences (HICSS)*, p. 6387–6396, 2020.

[10] J. Stirland, K. Jones, H. Janicke and T. Wu, "Developing Cyber Forensics for SCADA Industrial Control Systems," *Proceedings of the International Conference on Information Security and Cyber Forensics (InfoSec2014)*, pp. 98-111, 2014.

[11] T. Bihl, J. Schoenbeck, D. Steeneck and J. Jordan, "Easy and Efficient Hyperparameter Optimization to Address Some Artificial Intelligence "ilities"," *Hawaii International Conference on System Sciences (HICSS)*, pp. 943-952, 2020.

[12] S. Stone, M. Temple and R. Baldwin, "Detecting Anomalous PLC Behavior Using RF-Based Hilbert Transform Features and a Correlation-Based Verification Process," *Int'l J. Critical Infrastructure Protection*, vol. 9, pp. 41-51, 2015.

[13] A. C. Panchal, V. M. Khadse and P. N. Mahalle, "Security Issues in IIoT: A Comprehensive Survey of Attacks on IIoT and Its Countermeasures," *2018 IEEE Glob. Conf. Wirel. Comput. Netw.*, pp. 124-130, 2018.

[14] I. Yaqoob, I. A. T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar and S. Guizani, " Enabling Communication Technologies for Smart Cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 112-120, 2017.

[15] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Communications Magazine*, pp. 92-101, June 2010.

[16] ITU, *ITU-T G.9959: Short range narrow-band digital radio communication transceiver - PHY and MAC layer specifications*, Geneva, Switzerland: International Telecommunication Union, 2012.

[17] J. Sammons, *The basics of digital forensics: the primer for getting started in digital forensics*., Elsevier, 2012.

[18] M. Galeev, "Catching the Z-Wave," *Electronic Engineering Times India*, pp. 1-5, Oct. 2006.

[19] T. J. Bihl, K. W. Bauer and M. A. Temple, "Feature selection for RF fingerprinting with multiple discriminant analysis and using zigbee device emissions," *IEEE Trans. on Information Forensics and Security*, vol. 11, no. 8, pp. 1862-1874, 2016.

[20] C. K. Dubendorfer, B. W. Ramsey and M. A. Temple, "ZigBee device verification for securing industrial control and building automation systems," *Int. Conf. on Critical Infrastructure Protection (IFIP13)*, vol. 417, pp. 47-62, 2013.

[21] C. Rondeau, M. Temple and J. Betances, "Dimensional Reduction Analysis for Constellation-Based DNA Fingerprinting to Improve Industrial IoT Wireless Security," *Hawaii International Conference on System Sciences (HICSS)*, pp. 7126-7135, 2019.

[22] C. G. Looney, *Pattern Recognition Using Neural Networks*, Oxford University Press, 1997.

[23] T. Kohonen, J. Kangas, J. Laaksonen and K. Torkkola, "LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms," *Proceedings of the International Joint Conference on Neural Networks*, pp. 725-730, 1992.

[24] M. Kaden, M. Lange, D. Nebel, M. Riedel, T. Geweniger and T. Villmann, "Aspects in classification learning-Review of recent developments in Learning Vector Quantization," *Foundations of Computing and Decision Sciences*, vol. 39, no. 2, pp. 79-105, 2014.

[25] D. Nova and P. Estévez, "A review of learning vector quantization classifiers," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 511-524, 2014.

[26] A. S. Sato and K. Yamada, "Generalized learning vector quantization," in *Adv. in neural inform. processing sys*, 1995, pp. 423-429.

[27] B. Hammer and T. Villmann, "Generalized relevance learning vector quantization," *Neural Networks*, vol. 15, no. 8-9, pp. 1059-1068, 2002.

[28] D. DeSieno, "Adding a conscience to competitive learning," *Proc. IEEE Int'l Conf. Neural Networks*, 1988.

[29] P. Lorenzo, et al., "Particle swarm optimization for hyper-parameter selection in deep neural networks," *Proc. Genetic & Evolutionary Comput. Conf.*, pp. 481-488, 2017.

[30] J. Snoek, H. Larochelle and R. Adams, "Practical Bayesian optimization of machine learning algorithms," *Adv. in neural inform. processing sys.*, pp. 2951-2959, 2012.

[31] D. Montgomery, *Design and analysis of experiments*, John Wiley & sons., 2017.