Performance analysis of Extreme Learning Machine Classifiers on Radio Frequency Fingerprinting

Hüseyin PARMAKSIZ[1*] (ORCID: https://orcid.org/0000-0001-8455-5625), Cihan KARAKUZU[2] (ORCID: https://orcid.org/0000-0003-0569-098X).

Abstract— Internet of Things (IoT) is utilized in practically every industry. As IoT becomes more common, the number of wireless communication devices grows. The notion of security becomes more crucial as the number of devices and network grows. Due to welding constraints on IoT devices, the security can not be guaranteed. Radio frequency fingerprinting (RFF) methods, according to the literature, are utilized as an extra safety layer for wireless devices. Unique fingerprints due to the production defects of the devices are used to identify wireless devices for security purposes in order to avoid fraud or fraud attempts. In this study, a ready-made dataset, consisting of 3985 registered samples and transformed to nine extracted features, from four WiFi Access Point (AP) devices was used. Using this data set, classification performances of Extreme Learning Machine (ELM), Constrained ELMs (CELMs), and Meta-ELM techniques are examined. Considering the classification performance of the Meta-ELM algorithm, it is concluded that it can be used in RF fingerprinting research due to its superior performance. The use of Meta-ELM in multiple classification problems will be a novelty in the literature.

Index Terms— Internet of things, Radio frequency fingerprinting, Extreme learning machines, Fingerprint classification

I INTRODUCTION

With the extensive usage of smart devices in many different industries to improve people's standard of living, the network of IoT devices is fast expanding. Of course, because of the absence of security, it is easy to destroy smart gadgets. Security in these devices will not be viable in the near future due to a lack of hardware resources. Security is still an issue, resulting in security flaws in various gadgets. Although cryptographic methods are used for authentication, specifically to prevent attackers from accessing these devices, it is not a simple option for IoT owing to the computational complexity and scalability issues of these protocols [1]. RFF provides a physical layer-based security authentication environment, which is especially useful for low-resource devices. Because of the fault resulting from the device-based production stage, RFF is unique in this industry. Wireless signals generated by IoT devices are detected in the literature by software defined radio (SDR) devices [2]. RFF signals are detected and recorded using GNU Radio, Matlab, or other tools. The captured RFF signals are used to determine the properties. IoT devices are uniquely identified using classifier algorithms that utilise these attributes (fingerprints used in device identification).

Over the last several decades, many neural network architectures have been developed. The feed forward neural networks are among the most widely studied. It has been demonstrated that a multilayer feed-forward neural network with non-polynomial activation functions is capable of approximating any continuous function [3]. Researchers have extensively studied single hidden layer feed-forward neural networks (SLFNs) due to their simple modeling, relatively fast learning, and responsiveness. Gradient-based learning, Optimization-based learning, and Least Mean Square (LMS)-based learning are the three approaches for training SLFNs.

In the near past, Huang et al. [4] proposed ELM, a novel extremely fast learning model of SLFNs. ELM ensures the integrity of tasks such as classification, regression, semi-supervised, supervised, and unsupervised learning [5-7]. These benefits make ELM popular among researchers and engineers alike.

CELMs construct the parameters of hidden nodes in the standard ELM structure using a simple linear combination of sample vectors [8]. In CELM, the connection weights between the input layer and hidden neurons are drawn randomly from a constrained set of difference vectors of interclass samples, rather than an explicit set of random vectors [9].

Meta-ELM, can be considered as a hierarchical learning model. According to the problem handle training data set randomly reshuffled or not. And then, the training dataset is divided into subsets. Meta-ELM generates predictors on the subsets, and calculates predictor weights analytically, just like ELM [10].

Table 1 provides an overview of the features and classification techniques utilized in RFF applications. As can beseen from the table, ELM-based classifiers have not been run on this problem yet.

[1]*Corresponding author mail: huseyin.parmaksiz @bilecik.edu.tr (https://orcid.org/0000-0001-8455-5625)
Department of IT, Bilecik Şeyh Edebali University, Bilecik, TURKEY
[2] Second author mail: cihan.karakuzu@bilecik.edu.tr (https://orcid.org/0000-0003-0569-098X)
Department of Computer Engineering, Bilecik Şeyh Edebali University, Bilecik, TURKEY

II CLASSIFICATION ALGORITHMS

Numerous algorithms have been used for classification problems. A general categorization of these algorithms is given in Figure 1. In this section, the ELM-based algorithms discussed in this study will be briefly introduced based on the projection given in Table 1.

A ELM

Many engineering and science problems are solved using neural networks, and iterative algorithms are commonly used to train these networks. In feed-forward neural networks, iterative derivative-based algorithms are used to determine network parameters (thresholds and weights). New searches have begun as a result of the slow training time of derivative-based iterative algorithms. ELM, a learning algorithm designed for SLFN, overcomes this slowness. It was proposed in 2006 by Huang et al [4]. Although the ELM learning algorithm has a significant advantage in terms of training time, it does not perform as well in terms of generalization ability for small number of hidden neurons. ELM is a very fast learning method that has been proven by researchers to have high performance. However, its high generalization ability is based on a large number of hidden neurons, which is ineffective for real-time application response during testing.

When the number of neuron is bigger than the number of training samples (L>N), in any case, the linear system ($\boldsymbol{H\beta} = \mathbf{T}$) will have many solutions with zero error, resulting in overfitting. Eq. (1) can be used to train SLFN with L<N [4]. Training is done to minimize the cost function given in Eq. (2). T and t_j values in the equations represent the expected target in vector and matrice recpectively, and H represents the Huang matrice.

This process should be ended finding the specific $\hat{w}_i, \hat{b}_i, \hat{\beta} (i = 1, \dots, L)$ satisfying the condition given in Eq. (3).

$$|| \boldsymbol{H\beta} - \mathbf{T} ||_2 < \varepsilon, \tag{1}$$

$$C = \sum_{j=1}^{N} \left[\sum_{i=1}^{L} \beta_i G(w_i, b_i, x_j) - t_j \right]^2, \tag{2}$$

$$|| \boldsymbol{H}(\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_L, \hat{\boldsymbol{b}}_1, \dots, \hat{\boldsymbol{b}}_L)\hat{\boldsymbol{\beta}} - \boldsymbol{T}||_2 = \frac{min}{\mathbf{w_i}, \mathbf{b_i}, \boldsymbol{\beta}} || \boldsymbol{H}(\mathbf{w}_1, \dots, \mathbf{w}_L, \boldsymbol{b}_1, \dots, \boldsymbol{b}_L)\boldsymbol{\beta} - \boldsymbol{T} ||_2 \tag{3}$$

In contrast to traditional approximation theories, which require the adjustment of input weights and hidden layer biases, input weights and hidden layer biases can be randomly assigned if the activation function is infinitely differentiable, as rigorously demonstrated in [4]. To train an SLFN with fixed input weights ($\mathbf{w_i}$) and hidden layer biases($\mathbf{b_i}$), simply find the specific output weights ($\boldsymbol{\beta}$) such that

$$|| \boldsymbol{H\hat{\beta}} - \boldsymbol{T} ||_2 = \frac{\boldsymbol{min}}{\boldsymbol{\beta}} || \boldsymbol{H\beta} - \boldsymbol{T} ||_2 \tag{4}$$

ELM learns the output weight $\boldsymbol{\beta}$ by minimizing the cost function using Eq. (5), which is equivalent to determining the $\hat{\boldsymbol{\beta}}$ in Eq. (4).

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{5}$$

TABLE 1
RFF Features and Classification Algorithms.

Year/ Ref.	Feature/Method	Devices	Classification
2007 [11]	instantaneous attributes of the signals, principal component analysis (PCA features)	RF waveforms from WiFi devices	Probabilistic neural network (PNN)
2008 [12]	IQoffset, frequency error, phase and magnitude error, sync correlation.	802.11 NICs	SVM& k-NN
2009 [13]	Dual-tree complex wavelet transform	Wi-Fi 2 cards	Fisher-based MDA
2010 [14]	Instantaneous phase,amplitude and frequency (statistical features)	IoT – smart cards	Multiple Discriminant Analysis /Maximum Likelihood (MDA/ML)
2014 [15]	Clock Offset Kalman filters	Wireless systems	Measurement-hypothesis-filtering (MHF)
2017 [16]	instantaneous amplitude responses (Amplitude features) and their dimensionally reduced forms obtained by using (PCA features)	transmitters	PNN classifier outperforms the kNN
2017 [17]	Empirical Mode Decomposition in SEI Wigner-Ville distribution (WVD)	Mobile phones & WLAN cards	SVM
2019 [18]	Three-stage wavelet decomposition.	micro-UAV controllers	k-NN, SVM, DA, neural networks
2020 [19]	Time-domain RF signal.	Wi-Fi & ADS-B devices	CNN
2022 our study	RF signal statistical features.	Wi-Fi 2	ELM, CELMs, Meta-ELM

Where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H [20]. The solution $\hat{\boldsymbol{\beta}}$ defined in Eq. (5) is one of the least squares solutions (LSS) of linear system ($\boldsymbol{H\beta} = \mathbf{T}$), with the lowest norm among all solutions. According to [4], $\hat{\boldsymbol{\beta}}$ not only minimizes training error but also has the smallest weight magnitude. As a result, $\hat{\boldsymbol{\beta}}$ deserves to have the best generalization performance among all other LSS. In ELM, the orthogonal projection method can be used effectively: $\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\mathbf{H}^{\mathrm{T}}$ if $\mathbf{H}^{\mathrm{T}}\boldsymbol{H}$ is nonsingular, or $\mathbf{H}^{\dagger} = \mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}$ if $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is nonsingular [5]. The Moore-Penrose generalized inverse of H is calculated by singular value decomposition (SVD), iterative methods and orthogonalization methods [4].

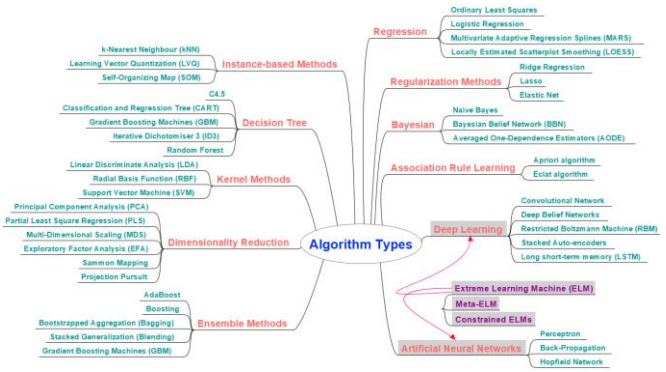

Figure 1 A General Categorization of Classification Algorithms and ELM Algorithms.

B Constrained ELMs

CELM is a single hidden layer feed-forward neural network based on ELM. The CELM generates its random weights from a smaller space than the ELM, by replacing completely random weight vectors with ones drawn at random from the set of difference vectors of between-classes samples. CELM's main contribution is that it incorporates sample distribution prior to the construction of the hidden layer, resulting in better discriminative feature mapping. The efficient use of hidden nodes in ELM is greatly aided by effective feature mapping. As a result, CELM is better suited for discriminatory tasks [8]. CELM has some issues like ELM. An overfitting problem occurs when the number of hidden nodes is excessively large (for example, 10K). It is suggested that this can be solved by including method in [21]. In addition, as the number of hidden nodes exceeds 10K, the generation of difference vectors will slow down, thus affecting the running speed of the algorithm.

CELMs are used to choose parameters of hidden neurons at random based on sample distribution. In contrast to ELM, which selects them at random, CELMs select them at random from a constrained vector space containing some basic combinations of original sample vectors. CELMs outperform ELM-related methods [22], SVM-related methods [23], and the BP neural network in terms of generalization while retaining ELM's fast learning characteristics [9].

In practice, the hidden layer's regular term with output link weights is added to the optimization target to avoid the problem [25, 26 and 27]. The regularized ELM solution can be obtained as follows and λ refers to the regularization factor in Eq. (6):

$$\beta = \mathbf{H}^T(\mathrm{I}/\lambda + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T} \tag{6}$$

The codes of CELMs used for comparison purposes in this study were obtained from the link "https://github.com/HuseyinPARMAKSIZ/Constrained-ELMs". Reference [8] contains the mathematical equations and algorithms of CELMs classifiers used in the study.

C Meta-ELM

Meta-ELM is a learning algorithm that combines a group of traditional ELM. It can be seen as an ensemble of standard ELMs. Meta-learning is a general technique for combining the outcomes of multiple learners, and it is loosely defined as learning from information generated by a learner(s). As shown in Figure 2, a Meta-ELM network model consists of a meta-approachist coupling structure called a meta-learner, taking the output of the baseline ELMs trained with subset of data as input. Each baseline ELM feeds the meta-learner as a baseline estimator, and the meta-approacher's parameters are deter-mined using all of the training data.

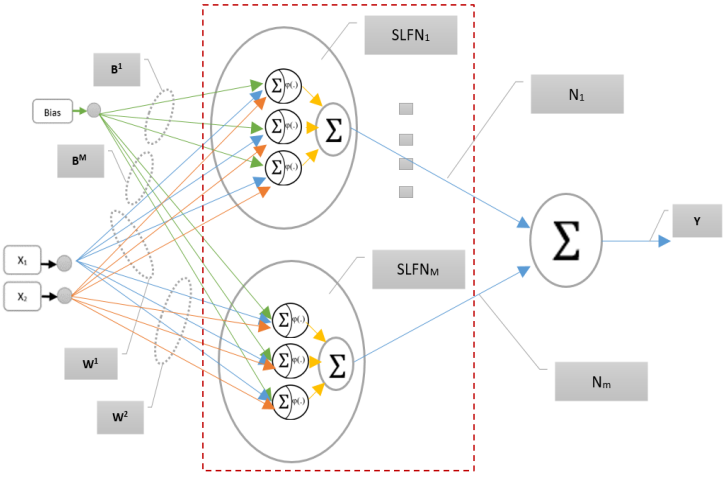

Figure 2 The Meta-ELM Architecture with 2 group (M=2) and 3 neuron (N=3).

Meta-ELM should minimize the cost function in Eq. (7):

$$C = \sum_{i=1}^{N}\left[\sum_{m=1}^{M}\beta_m(\mathbf{x}_i)\mathrm{ELM}_m(\mathbf{x}_i) - t_i\right]^2 \tag{7}$$

Where N denotes the number of training data, M denotes the number of base ELMs in the Meta-ELM model, $\mathrm{ELM}_m(\mathbf{x}_i)$ denotes the output of the mth base ELM given input$\mathbf{x}_i$, and β_m denotes the weight for the mth base ELM.

The Meta-ELM model's output function can be defined as Eq. (8):

$$f(x) = h(\mathrm{x})\,.\,\beta, where\ \boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{8}$$

III RFF and Dataset

The wireless RF given in Figure 3, which is part of the natural electromagnetic radiation spectrum, is between 3 kHz and 300 GHz frequency values. The spectrum used by wireless systems such as cell phones, radio and television broadcasts is in the critical frequency range. This spectrum covers frequencies in the [225 MHz to 3.7 GHz] range.

RF fingerprinting works similarly to how a listener can identify a speaker based on natural variations and characteristics of the voice. By extracting the time domain and frequency domain characteristics of the signal during operation, an RF fingerprint can automatically identify different wireless devices in the field [24].

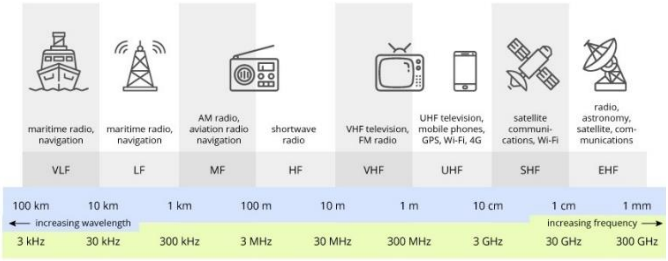

Figure 3 Wireless Radio Frequencies.

Devices may have uncontrollable random physical changes, randomness, and uniqueness that promotes non-repeatability during the manufacturing stage. In the literature, these characteristics are referred to as RF fingerprints. The RF fingerprint can be used as a separate physical layer security.

In the military, RF fingerprinting is routinely employed to track radars. RF fingerprinting has recently been used to identify and authenticate wireless devices. Because wireless is widespread in most IoT devices, it is also suitable for usage in this industry. In general, the RF fingerprint structure operates by sampling a radio signal from an emitting device. To avoid noise or other channel aberrations in the signal, signal preprocessing might be performed. The characteristics needed to differentiate the signal are then retrieved from the radio signal. These characteristics are then sent to a classifier, which specifically connects the radio signal to a certain device.

The data set (Ready9Ftrvete.mat file) used in the classification algorithm performance tests is available online in the "https://github.com/HuseyinPARMAKSIZ/RF-Ready-Dataset/" directory. This Wi-Fi fingerprint dataset was created using feature extraction methods on raw signals collected from four access points. In addition, features can be extracted from RF signals by applying statistical methods (such as skewness, kurtosis and variance) to the phase, frequency and amplitude information of the signal.

The signals recorded by Justice Owusu Agyemang are available at “https://github.com/jayluxferro/RFF”, as are various feature extraction methods from the raw format signal. We are grateful to him for granting us permission to use sample ready signals in our research.

Output classes are the number of related AP used for capturing RF signals. They are all numbered as (0-3) in data set. We used one-hot encoding for the output class. These numbers represent the classes of APs and are named Label in the dataset. 9 features were extracted from raw signals to be used in classification. These are "Mean, GMean, HMean, Median, MedianL, MedianH, MedianG, Variance and Stdev", respectively. There are 3985 records in this dataset. The training dataset contains 70% (2790 records) and the test dataset contains 30% (1195 records). Table 2 contains some sample data from the Wi-Fi RF dataset.

IV EXPERIMENTAL RESULTS

In the literature, as can be seen Fig. 1, many algorithms are used for classification purposes. In recent years, Deep Learning has been successfully applied to detect and classify RF signals. In this study, ELM-based algorithms, which have not encountered in RF fingerprint studies in the literature, were used for classification. Experiments was carried out in Matlab program running on an Elitebook with Windows 11 Pro 22H2 version operating system, Intel(R) Core(TM) i5-8250U CPU @1.60GHz processor and 32GB ram.

TABLE 2
Wi-Fi RF Dataset Sample Data.

Label	mean	geo-metric_mean	harmonic_mean	median	median_low	median_high	median_grouped	variance	stdev
0	12.23 0462 8658 941	12.18 5875 1284 791	12.14 1539 3845 330	12.08 8678 8922 157	12.08 8678 8922 157	12.08 8678 8922 157	12.08 8678 8922 157	1.144 27773 30437 8	1.069 70918 15272 9
0	10.97 1707 4806 488	10.91 0203 8384 247	10.84 9597 0933 690	10.95 3534 5628 792	10.95 3534 5628 792	10.95 3534 5628 792	10.95 3534 5628 792	1.397 60158 46782 3	1.182 20200 67138 4
1	9.732 1780 8461 114	9.711 8099 2418 995	9.691 2836 2967 730	9.822 4015 0579 026	9.788 8089 5444 767	9.855 9940 5713 284	9.355 9940 5713 284	0.436 50843 29604 28	0.660 68784 83523 28
1	10.31 5376 2115 175	10.30 5587 7068 646	10.29 5621 9061 378	10.39 6342 8928 298	10.35 0183 6900 156	10.44 2502 0956 439	9.942 5020 9564 394	0.220 39776 67791 53	0.469 46540 53060 28
2	13.64 7861 7912 927	12.85 0881 9577 644	11.98 1319 2831 179	14.04 3839 2922 497	13.81 9412 3981 631	14.26 8266 1863 363	13.76 8266 1863 363	19.59 20774 30816 8	4.426 29387 08152 6
2	9.335 4916 9900 563	7.571 5546 0629 170	6.164 5096 7083 778	8.265 1226 7002 011	8.058 5497 5489 044	8.471 6955 8514 977	7.971 6955 8514 977	33.37 05033 11649 4	5.776 72080 95639 7
3	15.30 1793 9982 726	14.55 2445 4905 452	13.57 2806 0531 529	16.77 5496 8395 464	16.77 5496 8395 464	16.77 5496 8395 464	16.77 5496 8395 464	18.40 03669 92318 6	4.289 56489 54548 5
3	14.53 9089 7135 477	13.77 1323 9019 030	12.76 1360 3584 628	14.90 3692 0871 237	14.90 3692 0871 237	14.90 3692 0871 237	14.90 3692 0871 237	18.07 28294 37300 5	4.251 21505 42286 8

A On the Meta-ELM Classification Performance

Classification performance of Meta-ELM is given with the confusion matrices given in Figure 4 by using the data set whose internet link given in Section III. No data processing has been done on the data set, it is already a data set that has been processed and given as attributes. All attributes are given directly as Meta-ELM inputs.

When the results in Table 3 are examined, 99.314% performance in the test set is obtained when M=100 and N=100. For

this reason, the values of M and N were chosen as 100 when constructing the confusion matrices. In the Figure 4, the matrices in the upper row are for the training set, and the ones in the lower row are for the test data set. The accuracy matrix's four classes represent APs. In the confusion matrix, the output class 1 value represents AP number 0. The value 2 denotes the AP number 1, the value 3 denotes the AP number 2 and the value 4 denotes the AP number 3. For example, of the 295 test set data belonging to the AP-0 class, 294 of them are correctly output class 1. But only 1 of them is incorrectly valued as output class 2. When the cases in other classes are analyzed similarly, the Meta-ELM (M, N=100) classifier algorithm we developed classifies the test set of the sample Wi-Fi RF fingerprint with 99.3% success. The performance of the same algorithm in the training phase is around 99.7%.

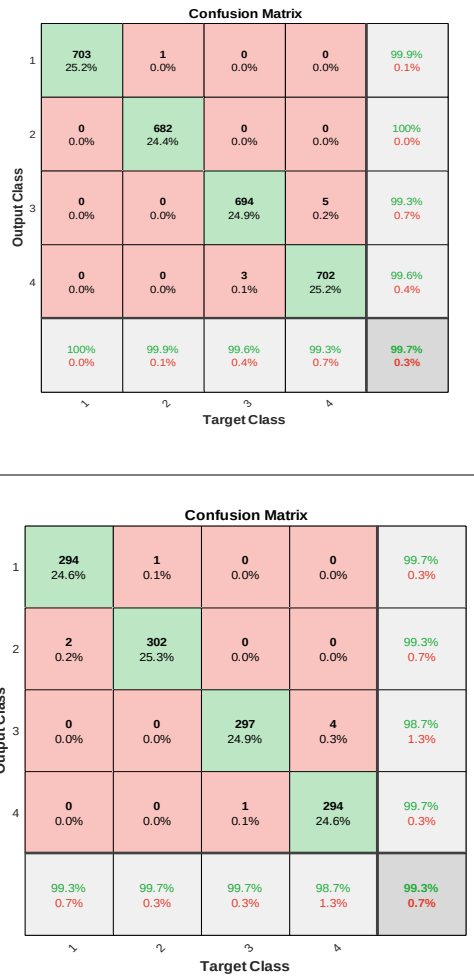

Figure 4 Confusion Matrix for Meta-ELM (up: train accuracy, down: test accuracy, M=100, N=100).

The M values used when comparing the classification accuracies of the Meta-ELM classifier are 10, 15, 25, 30, 50 and 100, and the N values are 2, 4, 8, 20, 50 and 100. After the system is trained with the training data, the accuracy values obtained for the test data are given in Table 3. The table shows the average values of the results obtained by running each algorithm separately five times. The best accuracy values for each M are given in bold, italic, and underline. In addition, for M=100, it is seen that the performance obtained for all N values is 98% and above. For N=100, when M=10, the accuracy is around 94%, while the accuracy increases as M increases. Within the framework of these results, it can be said that a very satisfactory performance of 99% can be obtained for M≥50 and N=100. As can be seen from the table, the testing time increased as the number of cells (N).

TABLE 3
Meta-ELM Classification Accuracy (M=number of groups, N = number of cells in each group, Acc=accuracy, time=testing time).

M=10	N=2	N=4	N=8	N=20	N=50	N=100
Acc	0.83582	0.88502	0.879	0.87063	0.8959	***0.94544***
time	3.9248	4.939	4.9003	6.5053	6.5273	8.6414
M=15	**N=2**	**N=4**	**N=8**	**N=20**	**N=50**	**N=100**
Acc	0.90946	0.91732	0.92921	0.88184	0.94711	***0.96402***
time	6.4189	6.7	7.8039	9.112	10.648	13.045
M=25	**N=2**	**N=4**	**N=8**	**N=20**	**N=50**	**N=100**
Acc	0.93406	0.95548	0.94745	0.94444	0.97272	***0.98209***
time	9.2511	9.5882	12.029	14.992	17.385	23.95
M=30	**N=2**	**N=4**	**N=8**	**N=20**	**N=50**	**N=100**
Acc	0.94996	0.95598	0.96033	0.9513	0.97958	***0.98343***
time	12.848	13.701	14.128	17.659	22.132	27.594
M=50	**N=2**	**N=4**	**N=8**	**N=20**	**N=50**	**N=100**
Acc	0.97891	0.97992	0.97774	0.97941	0.98929	***0.9913***
time	40.988	43.552	45.682	48.423	54.373	58.17
M=100	**N=2**	**N=4**	**N=8**	**N=20**	**N=50**	**N=100**
Acc	0.98611	0.98728	0.98611	0.99063	0.99213	***0.99314***
time	139	131.81	142.18	153.34	151.57	177.31

C Performance Comparison of ELM Based Some Algorithms

CELMs give improved accuracy performance over the basic ELM [8]. We examined the performance of a robust classifier (Meta-ELM) in our study by comparing different ELM kinds that have demonstrated their performance in the literature, because Meta-ELM fundamentally evolved with the advancements made on ELM. Based on this analysis, in experimental studies where we compared Meta-ELM with other ELM types, the lowest limit of M=10 groups was used. The average of accuracy values were calculated after the algorithms were ran five times. 30% of the data set consisting of 3985 records in total was tested with approximately 1195 records as the test set. The activation function *logsig* is used in all classifier algorithms. When a sigmoid additive node is used in the hidden layer of a case study in the literature, a different value in the form of multiples of 2 for regularizatin factor was used for 1000 cells in the classification with ELM in multi-class datasets [5]. The regu-

larizatin factor is usually determined empirically and varies depending on the problem and data set under consideration. In this study, this parameters has been determined using an analytical approach over training input matrice (X) as given Eq (9).

$$\boldsymbol{\lambda} = \max(\mathrm{eig}(\mathbf{X}^{\mathbf{T}}\mathbf{X})) \quad (9)$$

Table 4 compares the performance of the test data with the trained ELM, CELM, and Meta-ELM algorithms. In all tables in this paper, algorithm execution times for the test set are given in milliseconds. The test time increases as the number of cells (N) increases, as shown in the table. When N=2500, Meta-ELM and other constrained ELMs performed best, with the exception of DELM. ELM, on the other hand, performed best when N=500, and DELM performed best when N=1500. To emphasize the highest performance values, they are bolded, italic and underlined.

TABLE 4
Meta-ELM, ELM and CELMs Classifiers Accuracy
(M=number of groups, N = number of hidden nodes, Acc=accuracy, time=testing time).

Algorithm	N=100	N=500	N=1000	N=1500	N=2500
ELM Acc	0.9719	***0.9903***	0.9858	0.9846	0.9831
time	3	7.8	12.7	22.2	33.6
CELM Acc	0.9764	0.9863	0.9869	0.9871	***0.9878***
time	3	9.3	16.2	24.6	40.1
DELM Acc	0.9801	0.9854	0.9864	***0.9866***	0.9851
time	3.1	9.4	15.7	25.2	40.8
MELM Acc	0.9779	0.9836	0.9861	0.9869	***0.9871***
time	2.7	9.6	15.9	25.5	45.5
CSELM Acc	0.9680	0.9736	0.9746	0.9748	***0.9749***
time	2.6	9.2	14.9	27.9	40.1
RSELM Acc	0.9789	0.9871	0.9883	0.9885	***0.9896***
time	2.9	9.4	14.7	25.1	45.8
Meta-ELM M=25	N=4	N=20	N=40	N=60	N=100
Acc	0.95548	0.94444	0.96586	0.97623	***0.98209***
time	9.5882	14.992	16.71	21.337	23.95
Meta-ELM M=50	N=2	N=10	N=20	N=30	N=50
Acc	0.97891	0.93824	0.97941	0.98243	***0.98929***
time	40.988	46.434	48.423	51.373	56.373
Meta-ELM M=100	N=1	N=5	N=10	N=15	N=25
Acc	0.94494	0.98577	0.97031	0.98728	***0.99381***
time	90.176	132.66	136.64	138.32	151.08

In all ELM algorithms N represents the number of neurons in the hidden layer (sigmoid additive node). Since the Meta-ELM contains a group of ELMs, its total total number of cells is determined by the product of the number of groups (M) and the number of cells in each group (N). For this reason, M and N in Meta-ELM were adjusted so that equals the number of cells used in other constructs. As can be seen in Table 4, 100 neurons were used for Meta-ELM with M=25 and N=4, M=50 and N=2, M=100 and N=1. When M=100 and N=25, the highest performance of 0.99381% in the test set is obtained when M=100 and N=25 (2500 neurons in total).

V CONCLUSION

The performance of various ELM types has been evaluated for RF fingerprinting classification in this study. Furthermore, in the literature, the Meta-ELM algorithm is commonly used in regression problems, here, it has been adapted as classifier and used. In this regard, the presented research is unique and contributes the related area. In addition, it has achieved absolute performance accuracy compared to CELMs. Meta-ELM's ability to classify in RF fingerprints show that its usage in the future studies may get advantages. Furthermore, the Meta-ELM algorithm will bring new ideas to the literature in a variety of classification problems.

ACKNOWLEDGMENT

This study is supported by the project numbered 2021-01.BŞEÜ.01-01 within the scope of Bilecik Şeyh Edebali University Scientific Research Projects.

REFERENCES

[1] D. Nouichi, M. Abdelsalam, Q. Nasir, and S. Abbas, "IoT devices security using RF fingerprinting," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, 2019, pp. 1-7.

[2] H. Parmaksız and C. Karakuzu, "A Review of Recent Developments on Secure Authentication Using RF Fingerprints Techniques," *Sakarya University Journal of Computer and Information Sciences,* vol. 5, pp. -, 2022 (article in press).

[3] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural networks,* vol. 6, pp. 861-867, 1993.

[4] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing,* vol. 70, pp. 489-501, 2006.

[5] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 42, pp. 513-529, 2011.

[6] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE transactions on cybernetics,* vol. 44, pp. 2405-2417, 2014.

[7] C. Karakuzu, "Performance Comparison of a Neural Network and a Fuzzy Network Trained by ELM for Dynamic System Identification Problems," in *2nd International Congress on Engineering and Arthitecture (ENAR)*, 2019, pp. 22-24.

[8] W. Zhu, J. Miao, and L. Qing, "Constrained extreme learning machines: A study on classification cases," *arXiv preprint arXiv:1501.06115,* 2015.

[9] W. Zhu, J. Miao, and L. Qing, "Constrained extreme learning machine: a novel highly discriminative random feedforward neural

network," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 800-807.

[10] S. Liao and C. Feng, "Meta-ELM: ELM with ELM hidden nodes," *Neurocomputing,* vol. 128, pp. 81-87, 2014.

[11] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *Canadian Journal of Electrical and Computer Engineering,* vol. 32, pp. 27-33, 2007.

[12] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, 2008, pp. 116-127.

[13] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *Journal of Communications and Networks,* vol. 11, pp. 544-555, 2009.

[14] W. E. Cobb, E. W. Garcia, M. A. Temple, R. O. Baldwin, and Y. C. Kim, "Physical layer identification of embedded devices using RF-DNA fingerprinting," in *2010-Milcom 2010 Military Communications Conference*, 2010, pp. 2168-2173.

[15] M. M. U. Rahman, A. Yasmeen, and J. Gross, "Phy layer authentication via drifting oscillators," in *2014 IEEE Global Communications Conference*, 2014, pp. 716-721.

[16] S. Taşcıoğlu, M. Köse, and Z. Telatar, "Effect of sampling rate on transient based RF fingerprinting," in *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, 2017, pp. 1156-1160.

[17] J.-H. Liang, Z.-T. Huang, and Z.-W. Li, "Method of empirical mode decomposition in specific emitter identification," *Wireless Personal Communications,* vol. 96, pp. 2447-2461, 2017.

[18] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-UAV detection and classification from RF fingerprints using machine learning techniques," in *2019 IEEE Aerospace Conference*, 2019, pp. 1-13.

[19] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe*, et al.*, "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine,* vol. 3, pp. 50-57, 2020.

[20] C. R. Rao and S. K. Mitra, "Generalized inverse of a matrix and its applications," in *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability*, 1972, pp. 601-620.

[21] W. Zhu, J. Miao, and L. Qing, "Extreme support vector regression," in *Extreme Learning Machines 2013: Algorithms and Applications*, ed: Springer, 2014, pp. 25-34.

[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[23] O. Vinyals, Y. Jia, L. Deng, and T. Darrell, "Learning with recursive perceptual representations," *Advances in neural information processing systems,* vol. 25, 2012.

[24] T.-Y. Lin, C.-M. Lai, and C.-W. Chen, "Using SDR Platform to Extract the RF Fingerprint of the Wireless Devices for Device Identification," in *CS & IT Conference Proceedings*, 2020.

[25] G.-B. Huang, X. Ding and H. Zhou, “Optimization method based extreme learning machine for classification,” Neurocomputing, vol. 74(1), pp. 155-163, 2010.

[26] W. Zhu, J. Miao and L. Qing, “Robust regression with extreme support vectors,” pattern recognition letters, vol. 45, pp. 205-210, 2014.

[27] Q. Liu, Q. He and Z. Shi, “Extreme support vector machine classifier,” In Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, pp. 222-233, 2008.

Hüseyin PARMAKSIZ is a PhD student at Bilecik Şeyh Edebali University in Electronics and Computer Engineering. At the same time, he is a lecturer in the Network and System Branch Directorate of the relevant university's IT department and the head of the intervention in Cyber events. Linux, Open Infrastructure, Open-Source Software (OSS), Internet of Things (IoT), Security, Radio Frequency Fingerprinting, Extreme Learning Machines, and Heuristic algorithms are some of his research interests.

Cihan KARAKUZU is full professor at the Department of Computer Engineering, Faculty of Engineering, Bilecik Şeyh Edebali University, Bilecik, Turkey. His research interests include intelligent and artificial learning systems, fuzzy and neuro-fuzzy systems, data-driven models, system identification and modeling and meta-heuristic algorithms.