Machine Learning for QoS-Aware Fairness of a D2D Network

Xian Liu
Department of Systems Engineering
University of Arkansas at Little Rock
Little Rock, USA
xxliu@ualr.edu

Changcheng Huang
Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
huang@sce.carleton.ca

***Abstract*—In the quickly changing environment such as IoT, it is highly desirable to design a QoS-aware strategy to allocate the transmission power. In this paper, we apply the machine learning (ML) methodology to solve such a problem for a D2D network where the nodes are distributed following the conditional Poisson point process (PPP). The training is conducted in the feed-forward neural network (FNN).**

Keywords—Deep learning, machine learning, neural network, optimization, wireless networks.

I. INTRODUCTION

Device-to-device (D2D) *networking* is one of the representative wireless communication architectures in the current 4G and emerging 5G wireless communications. In particular, the D2D paradigm plays an important role in the *Internet of Things* (IoT). In practice, the D2D paradigm commonly coexists with the macro cellular network. Accordingly, there are two operating modes for the D2D communications: *underlay* and *overlay*. The underlay mode allows the D2D paradigm to share some spectra with the cellular network. The primary advantage of underlay is the spectrum efficiency, while the main disadvantage is the increased management complexity, since the cellular users should have the priority over the D2D users. On the other hand, the overlay mode reserves a small subset of spectra for the D2D users. Although the spectrum efficiency may not be as high as in underlay, the management complexity can be significantly reduced. Actually, if the overlay mode is implemented in a manner of time-multiplexing, periodically or statistically, the spectrum efficiency can be well improved. Nevertheless, in the overlay mode, several fundamental QoS issues must be well considered. For example, the *fairness* of resource allocation needs to be guaranteed within the D2D subnet. In most situations the task of allocating power to D2D transmitters is not trivial, due to the high density and mobility of D2D users. Moreover, the wireless channels are impacted by small and large scale fading as well as shadowing. The stochasticity is high.

In this paper, we develop an optimization model for a generic overlaid D2D subnet and solve it by the *machine learning* (ML) methodology [1]. Once the trained model is available, it can be used in a real-time mode which is highly desirable in the ever quickly changing environment such as IoT.

The rest of this paper is organized as follows. In Section II, the system model is described and the problem is formulated in a standard pattern. Considering that the presented strategy is a relative rare approach in the context of wireless communications, in Section III, we provide some remarks on the nonlinear optimization and *neural networks* (NN). Then the D2D networks used for training are described in Section IV. In Section V, three sets of NN training results of are discussed. Finally, the conclusion is put into Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first describe the system model of the D2D subnet, then propose an optimization model for its power allocation problem.

A. System Model

We consider a D2D subnet deployed in a cell represented by a disc with radius R_c. This D2D subnet consists of N pairs of *transmitters* (Tx) and *receivers* (Rx). The index set of these pairs is denoted by $\mathrm{N}_D = \{1, 2, \ldots, N\}$. All of these N pairs share the same frequency band. The main notations are listed in Table I. Other notations will be defined in the relevant context. In Table I, the term "link ij" means the link from Tx i to Rx j. The term "channel gain" refers to the small-scale fading. The Rayleigh fading is adopted in the analysis throughout this paper. Thus the channel gains h_{kj} and h_{kk} follow the exponential distribution. The conventions commonly used in the literature of wireless communications are also adopted, such as unit mean and the i.i.d. condition. This way, the large-scale fading due to the path-loss will be described with a power-law term. To concentrate on the key concept, the shadowing effect is included in the large-scale fading with appropriately adjusted parameters. In the present model, half-duplex is assumed, i.e., a node cannot receive signals while simultaneously transmitting signals. For example, at a particular moment, the link from Tx 3 to Rx 7 is different than the link from Tx 7 to Rx 3. This implies that, in general, it is not necessarily to have $b_{jk} = b_{kj}$, $b \in \{h, r, \alpha\}$. Also, $r_{kk} \neq 0$, since it represents the distance from Tx k to Rx k. σ_k^2 characterizes the *additive white Gaussian noise* (AWGN). With these elaborations, the *signal-to-interference-plus-noise ratio* (SINR) for the receiver k is expressed as:

$$u_k \overset{\Delta}{=} \frac{h_{kk}(P_k / r_{kk}^{\alpha_{kk}})}{\sigma_k^2 + \sum_{j=1, j\neq k}^{N} h_{jk}(P_j / r_{jk}^{\alpha_{jk}})}, \quad (k = 1, 2, \ldots, N) \tag{1}$$

978-1-7281-5207-3/20/$31.00 ©2020 IEEE

where usually $1.6 < \alpha_{jk} < 9$. Note that in (1) h_{jk}, h_{kk}, r_{jk}, and r_{kk} are *random variables* (RVs), while the variables P_j and P_k are the entities to be optimized (referred to as the *decision variables* in the optimization literature).

TABLE I. LIST OF MAIN NOTATIONS

$E(\tau)$	Mean of the random variable τ
h_{jk}	Channel gain of the interference link jk
h_{kk}	Channel gain of the desirable link paired with Tx k and Rx k
N	Number of Tx-Rx pairs
r_{jk}	The length of link jk
u_k	SINR of Rx k
P_k	Transmitter power of Tx k
α_{jk}	Path-loss exponent of link jk
σ_k^2	Noise power of Rx k

B. Problem Formulation

In the present work, we develop a model to optimally allocate the power for all D2D transmitters. There are several ways to characterize the merit of scheme. A variety of examples can be found in the literature, e.g., [2, Ch.6] and [3]. Here we adopt the proportional fairness as the objective function. The problem is formulated as follows:

(PF1)

$$\text{maximize } z = \left\{ \prod_{k=1}^{N} \log_2(1+u_k) \right\}^{1/N}; \quad (2)$$

$$\text{subject to } u_k \geq u_{k,\min}; \quad (3)$$

$$0 \leq P_k \leq P_{k,\max}; \quad (4)$$

$$(k = 1,2,...,N)$$

where u_k is defined in (1).

Remark 1: The objective function in PF1 is in terms of the *geometric average*.

Remark 2: In (1) and PF1, P_j (P_k) are decision variables, while others are parameters.

Remark 3: PF1 characterizes a single frequency band only. Due to the orthogonality implied by overlay, it is straightforward to apply PF1 to the paradigm where more than one frequency bands are allocated to the D2D users.

Remark 4: PF1 is a QoS-aware model due to the constraint set (3). According to (1), the constraint (3) appears to be highly nonlinear:

$$\frac{h_{kk}(P_k / r_{kk}^{\alpha_{kk}})}{\sigma_k^2 + \sum_{j=1, j\neq k}^{N} h_{jk}(P_j / r_{jk}^{\alpha_{jk}})} \geq u_{k,\min}. \quad (5)$$

This intact fractional formulation could be directly dealt with by most standard numerical optimization solvers, such as those in Matlab. However, it is easy to transfer (5) into an equivalent linear form:

$$h_{kk}(P_k / r_{kk}^{\alpha_{kk}}) - u_{k,\min} \sum_{j=1, j\neq k}^{N} h_{jk}(P_j / r_{jk}^{\alpha_{jk}}) \geq u_{k,\min}\sigma_k^2. \quad (6)$$

In the numerical experiments, we found that the behavior of the standard numerical optimization solvers with (6) is much better than using (5) as-is. This example shows the significant difference between pure mathematics and numerical mathematics. We also remark that the performance of numerical optimization can be further improved by using a normalized version of (6):

$$\frac{h_{kk}P_k}{u_{k,\min}\sigma_k^2 r_{kk}^{\alpha_{kk}}} - \frac{1}{\sigma_k^2} \sum_{\substack{j=1 \\ j\neq k}}^{N} \frac{h_{jk}P_j}{r_{jk}^{\alpha_{jk}}} \geq 1. \quad (7)$$

III. ESSENTIALS OF NEURAL NETWORK

Mathematically, the PF1 model described in the preceding section is a constrained *nonlinear programming* (NLP) problem. It could be solved by several numerical algorithms. To promote the real-time mode, in the present work, we investigate the feasibility of the machine learning (ML) methodology. The modern ML discipline is built upon the neural networks (NN). In general, NN can perform the *supervised training* or *unsupervised training*. NN can also be applied to *pattern classification* or *function approximation*. For the present problem, we mainly exploit NN for function approximation and supervised training.

The standard NN consists of three processes: *training*, *validation*, and *testing*. In the training stage, a set of data samples are used to solve the unknowns (a,k,a, weights and biases) of NN. In the validation stage, the *hyper-parameters* (HPs) are tuned up. One of the example HPs is the polynomial degree in the NN involving nonlinear regression. In the testing stage, a set of new data samples are used to verify the established mapping by NN.

Two types of datasets are needed for the supervised training. Before training an NN, a standard NLP solver is used to obtain a set of optimal solutions. This dataset is used as the *target* dataset for the NN. The other dataset is the input to the NN, commonly formatted as an $R \times Q$ matrix, where R is the number of elements in the NN's input, while Q is the number of samples [4]. Corresponding the PF1 model, R is the number of total random channel gains, while Q is the number of total scenarios.

IV. DATASET GENERATOR

As addressed in the preceding section, the concerned NN problem falls into the category of function approximation and supervised training. Therefore, the dataset for training the NN must be provided. In the present work, we designed a dataset generator for this purpose.

A. Distribution of Transmitters

We consider a generic overlaid D2D subnet deployed in a region represented by a disc with radius R_c. The total number of transmitters (Tx) is N. These N transmitters are independently uniformly distributed in the disc. Thus the *probability density function* (PDF) is $1/(\pi R_c^2)$. Accordingly, in the polar coordinate system, the *cumulative distribution function* (CDF) is:

$$P_R(R \le r, \Theta \le \theta) = \int_0^{\theta} \int_0^{r} \frac{1}{\pi R_c^2} \rho d\rho d\beta = \frac{\theta r^2}{2\pi R_c^2}. \quad (8)$$

Due to the omni-directionality property, we have:

$$F_R(r) = P_R(R \le r) = \frac{r^2}{R_c^2}. \quad (9)$$

Following the *inverse function method* [5, Sec. 4.9.1], we set:

$$R = F_R^{-1}(U) = R_c \sqrt{U}, \quad (10)$$

where U is a random variable uniformly distributed in [0, 1]. Then the random variable R will follow the CDF in (9). Next, another uniform random variable Θ is generated over $[0,2\pi]$. Consequently, in the Cartesian coordinate system, we have:

$$X = R_c \sqrt{U} \cos\Theta, \quad (11)$$

$$Y = R_c \sqrt{U} \sin\Theta. \quad (12)$$

It is worth noting that the above model is equivalent to the homogeneous *Poisson point process* (PPP) conditioned that there are N points within a finite disc [6, Theorem 2.9].

B. Distribution of Receivers

For a particular Tx, there is one associated receiver (Rx). The distribution of Rx follows the *bipolar* distribution [6, Sec. 5.3], which is a popular scheme in small-scale and ad hoc wireless networks. In this type of deployment, there is a circle centered at each Tx with radius

$$b_0 = (0.5 - \varepsilon) \min_{j,k \in N_D} r_{jk}, \quad (13)$$

where $\varepsilon > 0$ is a small number to avoid an Rx is exactly at the midpoint between two Tx. As indicated in [6], several practical configurations can be easily derived from the bipolar model.

For the small-scale fading, we assume that its channel gains follow the exponential distribution with unit mean. A snap shot is shown in Fig. 1, where the corresponding *Voronoi tessellations* are also illustrated. The main parameters are listed in Tables II.

TABLE II. MAIN PARAMETERS IN TESTBED [7]

Parameter	Value
Disc radius	100 (m)
Path-loss exponent (α)	4
Maximum transmitter power $(P_{\max})$	0.1 (mW)
Noise variance (σ^2) at 1 MHz bandwidth	-143.97 (dBm)
Number of Tx-Rx pairs	10, 20
Number of scenarios	6800, 22110

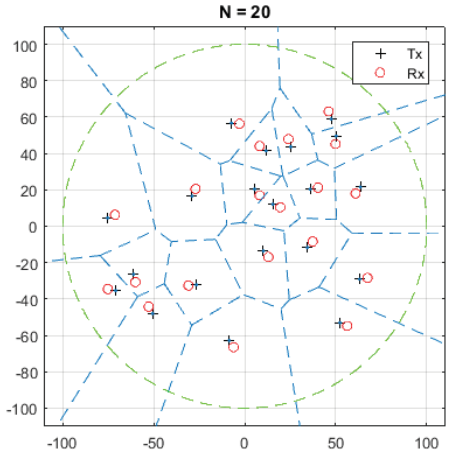

Fig. 1. A snapshot of D2D deployment.

V. NEURAL NETWORK TRAINING

In the present work, the training is conducted in the *feed-forward neural network* (FNN) [1, Ch. 6]. It was implemented in Matlab [4]. One of the axioms of NN is to staring the small architecture whenever possible. We have investigated several small architectures. In this paper, we present a design where the FNN consists of three hidden layers, with 50, 20, and 20 neurons, respectively. All simulations were conducted in a computer equipped with the 3.50 GHz CPU and 16 GB RAM.

A. Experiment 1

We started the training activity for a simplified version of PF1, where the QoS constraints (3) are relaxed. For the case of $N = 10$, the dimension of each single input vector is $N^2 = 100$. The total number of input vectors (samples) is chosen as 6800, which is slightly larger than the total unknown parameters of this FNN to mitigate over-fitting. Similarly, the number of target vectors is also 6800. They are generated by the NLP solver `fmincon` in Matlab. Three instances of the targets are presented in Table III, where the data are normalized to $P_{\max}$.

In the setup of NN, 70%, 20%, and 10% of samples are used as the datasets for training, validation, and testing, respectively. These ratios are based on typical recommendations in most NN monographs. A set of profiles is illustrated in Figs 2 to 4. Note that these profiles are recognized as standard benchmarks in the NN discipline. In Fig. 2, it is shown that the validation and test errors increases after the 30th epoch. We used the default threshold (6 epochs) as the stopping criterion. In Fig. 3, it is shown that a dominant portion of errors falls in two very narrow intervals and, actually, presents two "pulses". Finally, in Fig. 4, the scatter plots of all sub-datasets are illustrated. They are the linear fitting between the target data and the calculated data by

this FNN. In the case of perfect fitting, the value of index R should be 1. We obtained an overall value $R \approx 0.935$. This is mainly due to the relatively small size of the FNN.

TABLE III. EXAMPLES OF TARGET MATRIX

Row	Column 1	Column 1000	Column 3000
1	0.3786	0.1763	0.0370
2	0.8560	0.3904	0.2132
…	… …	… …	… …
10	0.1133	0.5501	0.1677

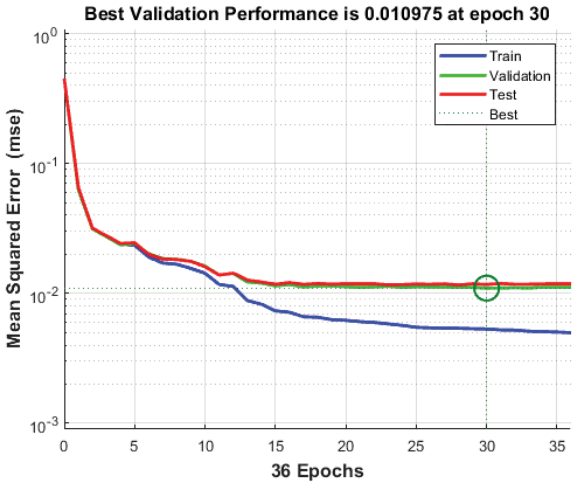

Fig. 2. MSE performance 1.

B. *Experiment 2*

In this experiment, we increased the number of D2D pairs to $N = 20$. Although this setup only doubled the previous case, the training time was significantly increased, partially due to the larger number of input vectors: 22110. With the same portions that 70%, 20%, and 10% of samples were used as the datasets for training, validation, and testing, respectively. The obtained results are illustrated in Figs. 5 and 6. In Fig. 5, it is shown that the validation and test errors keep decreasing through the 33rd epoch. In Fig. 6, the scatter plots of all sub-datasets are illustrated. We obtained an overall value $R \approx 0.879$. It is slightly lower than the case of $N = 10$. The Histogram of errors is omiited here due to the space limit.

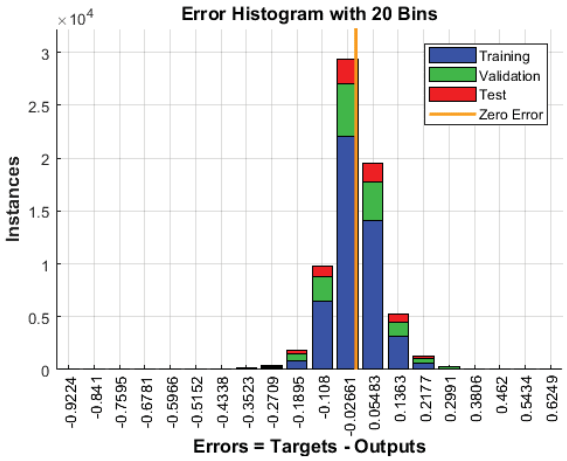

Fig. 3. Histogram of errors 1.

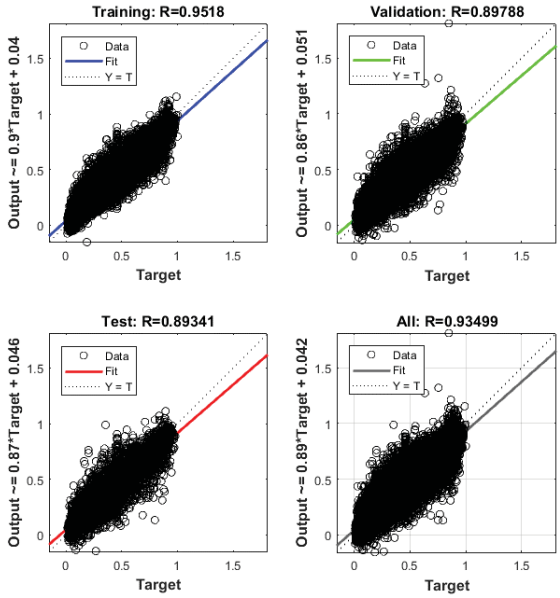

Fig. 4. Scatter plots 1.

C. *Experiment 3*

In this experiment, we included the QoS-aware constraints (3) in PF1. The number of D2D pairs was set to $N = 10$. The obtained results are illustrated in Figs. 7 to 8. In Fig. 7, it is shown that the validation and test errors increases after the 33^{ed} epoch. We firstly used the default threshold (6 epochs) as the stopping criterion. Then 20 epochs were used to see whether the performance could be improved. This change can be seen from Fig. 7, where the horizontal axis extends to 55 epochs. However, as shown in Fig.8, the value of index R is lower than previous examples. We conjecture that it was due to the relatively small size of NN. This is currently restricted by the available computational platform. We will conduct the training on the large NNs once the resource is upgraded in the near future.

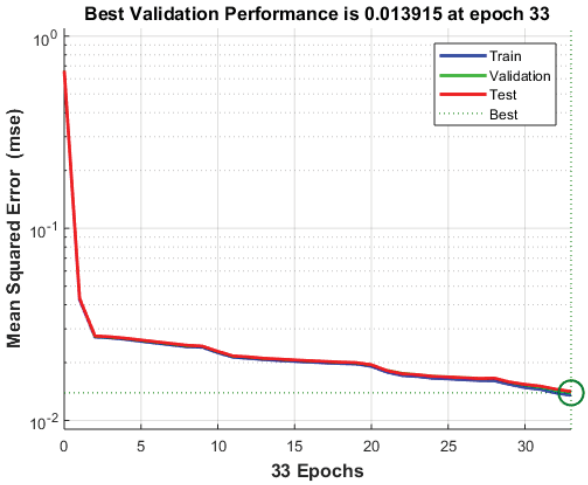

Fig. 5. MSE performance 2.

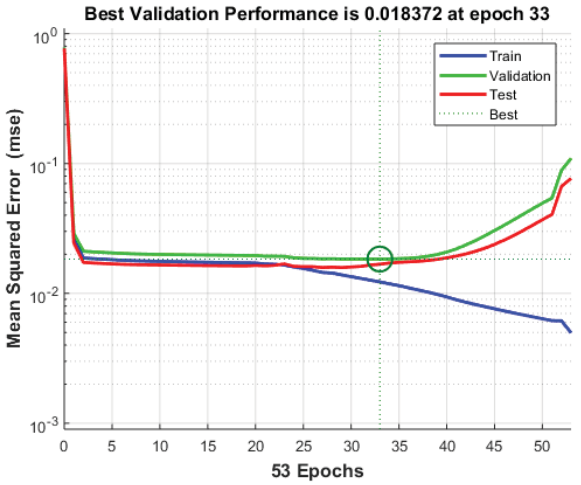

Fig. 7. MSE performance 3.

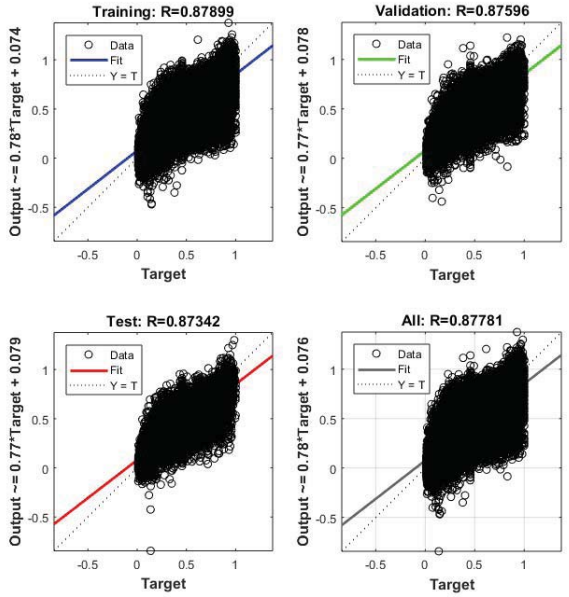

Fig. 6. Scatter plots 2.

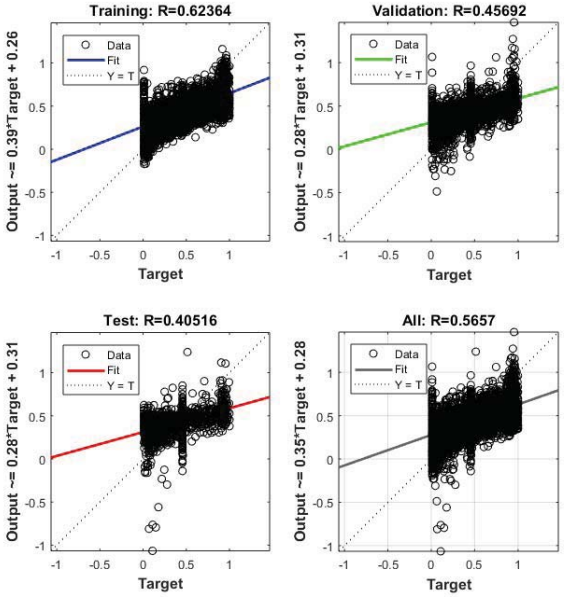

Fig. 8. Scatter plots 3.

VI. Conclusion

In this paper, we solve the power allocation problem for the overlaid D2D subnet with QoS requirements. We present a new approach to seek the fairness solutions by means of machine learning (ML). The power allocation with fading immunization is thus induced from the fitting function realized in ML. The training is conducted in a feed-forward neural network. The details of this network are given in the paper.

Considering that the presented strategy is a relative rare approach in the context of wireless communications, we also provided some remarks on the relation between the nonlinear optimization and ML. These remarks would be helpful to gain more key insights of the proposed approach. In the on-going research, we are making efforts to extend the present work to other types of communication networks.

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.

[2] S. Verdu, *Multiuser Detection*, Cambridge University Press, Cambridge, UK, 1998.

[3] Z-.Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, 2008.

[4] Neural Network Toolbox, Matlab, R2018a.

[5] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering* (3rd ed.), Pearson, Upper Saddle River, NJ, 2008.

[6] M. Haenggi, *Stochastic Geometry for Wireless Networks*, Cambridge University Press, 2012.

[7] N. Lee, X. Lin, J. G. Andrews, and R. W. Heath, "Power control for D2D underlaid cellular networks: modeling, algorithms, and analysis", *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 1, pp. 1-13, 2015.