A Hybrid Multiobjective RBF-PSO Method for Mitigating DoS Attacks in Named Data Networking

Amin Karami*, Manel Guerrero-Zapata

Computer Architecture Department (DAC), Universitat Politècnica de Catalunya (UPC), Campus Nord, C. Jordi Girona 1-3. 08034 Barcelona, Spain

Abstract

Named Data Networking (NDN) is a promising network architecture being considered as a possible replacement for the current IP-based (host-centric) Internet infrastructure. NDN can overcome the fundamental limitations of the current Internet, in particular, Denial-of-Service (DoS) attacks. However, NDN can be subject to new type of DoS attacks namely Interest flooding attacks and content poisoning. These types of attacks exploit key architectural features of NDN. This paper presents a new intelligent hybrid algorithm for proactive detection of DoS attacks and adaptive mitigation reaction in NDN. In the detection phase, a combination of multiobjective evolutionary optimization algorithm with PSO in the context of the RBF neural network has been applied in order to improve the accuracy of DoS attack prediction. Performance of the proposed hybrid approach is also evaluated successfully by some benchmark problems. In the adaptive reaction phase, we introduced a framework for mitigating DoS attacks based on the misbehaving type of network nodes. The evaluation through simulations shows that the proposed intelligent hybrid algorithm (proactive detection and adaptive reaction) can quickly and effectively respond and mitigate DoS attacks in adverse conditions in terms of the applied performance criteria.

Keywords: Named Data Networking, DoS attacks, Intelligent Hybrid Algorithm, RBF neural networks, Particle Swarm Optimization, NSGA II

1. Introduction

In recent years there have been several efforts to design new network architectures for a viable and vital replacement for the current IP-based Internet [1, 2, 3]. These new architectures are designed to better cope with the fundamental limitations of the Internet in supporting today's content-oriented services [4, 5]. In particular, there have been significant efforts regarding security, privacy, better mobility, scalability and efficient content distribution [6, 7]. Strong security has been one of the main design requirements for these architectures [8, 9]. Named Data Networking (NDN) [10] is one of these architectures as an ongoing research effort that aims to move the Internet into the future with a content-centric approach. NDN is a prominent example of Content-Centric Networking (CCN, also referred to as Information-Centric Networking (ICN) or Data-Centric Networking (DCN)) design. In CCN, content -rather than hosts, like in IP-based design- plays the central role in the communications [11, 12].

In contrast to today's Internet, a key goal of the NDN project is "security by design" [13, 14, 15]. Unlike the current Internet (host-based) approach in which security, integrity and trust should be provided in the communication channel, CCN secures content (information) itself and puts integrity and trust as the content properties [16, 17]. However, with this new paradigm, new kinds of attacks and anomalies -from Denial of Service (DoS) to privacy attacks- will arise [18, 19]. The big question is how resilient will this new NDN architecture be against DoS/DDoS attacks [15, 20]. An adversary can take advantage of two features unique to NDN namely Content Store (CS) and Pending Interest Table (PIT) to mount DoS/DDoS attacks specific to NDN such as Interest flooding attacks and content poisoning [20, 21].

*Corresponding author, Telephone: 0034-934011638

Email addresses: amin@ac.upc.edu (Amin Karami), guerrero@ac.upc.edu (Manel Guerrero-Zapata)

URL: http://personals.ac.upc.edu/amin (Amin Karami), http://personals.ac.upc.edu/guerrero (Manel Guerrero-Zapata)

The first goal of any protection scheme against DoS attack is the early detection (ideally long before the destructive traffic build-up) of its existence [22, 21]. In order to disarm DoS/DDoS attacks and any deviation, not only the detection of the malevolent behavior must be achieved, but the network traffic belonging to the attackers should be also blocked [23, 24, 25]. Thus, a predictor (detector) should take an appropriate action to thwart the attacks and should be able to adjust itself to the changing dynamics of the anomalies/attacks [20, 26]. In an attempt to tackle with the new kinds of DoS attacks and the threat of future unknown attacks and anomalies, many researchers have been developing intelligent learning techniques as a significant part of the current research on DoS attacks detection [16, 27]. The most popular approach for DOS/DDoS attacks prediction is using Artificial Neural Networks (ANNs) classification [28, 29, 30]. ANNs have become one of the most vital and valuable tools in solving many complex practical problems [31, 32], among which the Radial basis function (RBF) neural networks have been successfully applied for solving dynamic system problems, because they can predict the behavior directly from input/output data [33, 34, 35]. RBF networks have many remarkable characteristics, such as simple network structure, strong learning capacity, better approximation capacities and fast learning speed. The difficulty of applying the RBF networks is in network training which should select and estimate properly the input parameters including centers and widths of the basis functions and the neuron connection weights [32, 36, 37]. In order to find the most appropriate parameters, an optimization algorithm can be used [38, 39]. An optimization algorithm will attempt to find an optimal choice that satisfies defined constraints and make an optimization criterion (performance or cost index) maximize or minimize [38, 40]. Hence, to improve the prediction accuracy and robustness of the RBF network, network parameters (centers, widths and weights) should be simultaneously tuned [32]. Some of the existing algorithms to achieve that are given in [32, 36, 41, 42, 43]. Almost all algorithms compute the optimal estimation of the basis function centers by mean of error minimization, i.e., accuracy based on Mean-Square Error (MSE) [36, 43, 44, 45]. However, MSE is not suitable for determining the optimal position of basis function centers. Since the MSE decreases, the number of centers increases [46]. To accomplish this task, we develop our proactive detection algorithm for globally well-separating units' centers and their local optimization by MSE (decreasing the error caused by corresponding data points and their centers, separately). But the optimal placement and well-separated centers can increase MSE [47]. It is generally accepted that well-separated (external separation of) centers and their local optimization (internal homogeneity) are conflicting objectives [46, 48]. This trade-off is a well-known problem as the Multiobjective Optimization Problem (MOP) [49, 50, 51, 52]. This paper applies NSGA II (Non-dominated Sorting Genetic Algorithm) proposed by Deb et al. (2002) to solve this problem, as it has recently been frequently applied to various scenarios [53, 54, 55, 56]. On the other hand, for (near) optimal estimation and adjustment of two others RBF parameters (units' widths and output weights), we implement Particle Swarm Optimization (PSO) that favors global and local search of its interacting particles which has proved to be effective in finding the optimum in a search space [57, 58, 59].

When the DoS attacks by the proposed intelligent predictor are identified, the second phase (i.e., adaptive mitigation reaction) is triggered by enforcing explicit limitations against adversaries. The contribution of this paper is summarized in three objectives. The first objective of this paper is to develop an algorithm to resolve the hybrid learning problem of a RBF network using multiobjective optimization and particle swarm optimization to obtain a simple and more accurate RBF network-based classifier (predictor). The second objective is utilization of this optimized RBF network-based predictor in proactive detection of the DoS/DDoS attacks in NDN. The third objective is introducing a new algorithm to enable NDN routers to perform quickly and effectively adaptive reaction (recovery) from network problems, in order to keep track of legitimate data delivery performance and effectively shutting down malicious users' traffic.

There are three main advantages of the proposed prediction (classification) method; first, the proposed method can be applied to classification of any real-world problem; second, it gives better results in terms of the low misclassification, accuracy and robustness for some benchmark problems. And third, it provides a promising performance in prediction of DoS attacks in NDN. Moreover, the evaluation through simulations shows that the proposed intelligent hybrid algorithm (proactive detection and adaptive reaction) can quickly and effectively respond and mitigate DoS attacks in adverse conditions in terms of the applied performance criteria.

The rest of the paper is organized as follows. Section 2 describes the background materials of the NDN, and follows by the explanation of the related DoS attacks in Section 3. Section 4 provides the related work. Section 5 presents RBF neural networks. Section 6 describes the PSO algorithm. In Section 7, NSGA II is presented

in detail. Section 8 describes the proposed hybrid intelligent method. Section 9 examines the detection (classification) phase of the proposed hybrid algorithm on several benchmark problems. Evaluation environment is described in Section 10 in detail. Section 11 proposes our countermeasure including proactive detection and adaptive reaction mechanism against DoS attacks in NDN. Detailed analysis and discussions are explained in Section 12. Finally, Section 13 draws conclusions.

2. NDN overview

NDN is a novel next-generation Internet architecture based on the principle of CCN (Content-Centric Networking) paradigm, where contents are retrieved by their names instead of by the network addresses where they are hosted [60]. Data names in NDN are hierarchically structured, e.g., eighth fragment of a youtube video (file) would be named `/youtube/videos/A7m5I6n8kVUw/8`. NDN is one of the five NSF-sponsored Future Internet Architecture (FIA) [21]. It supports two types of messages: *Interest* and *content* (Fig. 1). A consumer asks for a content

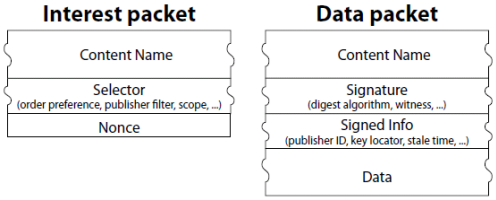


Figure 1: CCN packet types [12]

by routing an Interest request using name prefixes instead of today's IP prefixes. Interest packet is routed towards the location of the origin content where it has been published. Any router and middle node on the way checks its cache for matching copies of the requested content. If a cached copy of any piece of Interest request is found, it is returned to the requester along the path the request came from. On the way back, all the middle nodes store a copy of content in their caches to answer to probable same Interest requests from subsequent requests [16]. Each NDN router maintains three major data structures including *Forwarding Information Base* (FIB), *Pending Interest Table* (PIT) and *Content Store* (CS) or buffer memory. FIB is a lookup table used to determine interfaces for forwarding incoming Interests packets which contains [*name prefix* , *interface number*] entries. Another lookup table is PIT which contains outstanding entries [*Interest prefix* , *arrival interface*]. When a NDN router receives an Interest packet, it first checks its CS (cache). If there is no copy of the requested content, it looks up its PIT table. If the same name is already in the PIT and the arrival interface of the present Interest is already in the set of *arrival interface* of the corresponding PIT entry, the Interest is discarded. If a PIT entry for the same name exists, the router updates the PIT entry by adding a new arrival interface to the set. The Interest is not forwarded further. Otherwise, the router creates a new PIT entry and forwards the present Interest using its FIB [12, 61].

3. DoS attacks in NDN

The new variations of DoS attacks might be quite effective against NDN. An adversary can take advantage of two features unique in NDN routers as CS and PIT to mount DoS/DDoS attacks into NDN. There are two major categories of DoS attacks in NDN infrastructure [20, 62]:

1. *Interest Flooding Attack (IFA)*: It is partly due to the lack of authentication of Interest packets (source). Anyone can generate Interests packets and any middle router (node) only knows that a particular Interest packet entered on a specific interface.
2. *Content/Cache Poisoning*: The adversary tries to make routers forward and cache corrupted or fake data packets in order to prevent consumers from retrieving the original (legitimate) content.

3.1. Interest Flooding Attack

In this type of attack, the adversary (controlling a set of possibly geographically distributed zombies) generates a large number of Interest packets aiming to (1) overwhelm PIT table in routers in order to prevent legitimate users to satisfied their Interest packets and (2) swamp the target content producers [20]. There are three types of Interest flooding attacks, based on the type of content requested [20]:

1. *existing or static*: it is quiet limited since in-network content caching provides a built-in countermeasure. If several zombies from different paths generate large number of Interest packets for an existing content which settles in all intervening routes' caches, these Interest packets for the same content can not propagate to the producer(s) since they are satisfied by cached copies.
2. *dynamically-generated*: There is no benefits via caching copies. Since requested content is dynamic, all Interest packets are routed to content

producer(s), thus consuming bandwidth and router PIT table. Also, content producer might waste considerable computational resources due to the signing the content (per-packet operation) which is itself expensive.

3. *non-existent (unsatisfied Interests)*: Such Interest packets cannot be collapsed by routers, and are routed toward the content producer(s). This type of Interest packets take up space in router PIT table until they expire. A large number of non-existent Interest packets in PIT table lead to legitimate Interest packets being dropped in the network.

4. Related work

As a new Internet architecture proposal, there is very limited work recently regarding to mitigation of DoS/DDoS attacks in Named Data Networking. Gasti et al. [20] performed initial analysis of NDN's resilience to DoS attacks. This work identifies two new types of attacks specific to NDN (Interest flooding and content/cache poisoning) and discusses effects and potential countermeasures. However, the paper does not analyze DoS attacks and their countermeasures. Afanasyev et al. [15] presented three mitigation algorithms (token bucket with per interface fairness, satisfaction-based Interest acceptance and satisfaction-based pushback) that allow routers to exploit their state information to thwart Interest flooding attacks. Among these three mitigation algorithms, satisfaction-based pushback mechanism could effectively shut down malicious users while preventing legitimate users from service degradation. This work uses a simple and static attackers model (sending junk Interests as fast as possible), and it does not consider intermediate router's cache and always forwards all the way to the producer. Compagno et al. [21] introduced a framework for local and distributed Interest flooding attack mitigation, in particular, rapid generation of large numbers of Interest for non-existent contents that saturate the victim router's PIT. Authors simulated a simple attackers model, and their countermeasure has been able to use around 80-90% of the available bandwidth in the most cases during the attacks. Dai et al. [63] proposed Interest traceback as a counter measure against NDN DDoS attacks, which traces back to the originator of the attacking Interest packets. In this paper, when PIT exceeds its threshold, Interest traceback is triggered. This method responds to the attack by generating spoofed Data packets to satisfy the long-unsatisfied Interest packets in the PIT by tracing back to the Interest originators. This method is not proactive, makes overhead in the network by increasing of made spoofed contents. It leads to middle routers cache bogus contents. This paper also assumes that the long-unsatisfied Interests in the PIT is adversary and others unsatisfied Interest are normal usages. Another shortcoming of this method is that the router drops the incoming packet rate of the interface which has too many long-unsatisfied Interest packets. As a result of this independent decision, the probability of legitimate Interests being forwarded decreases rapidly as the number of hops between the content requester and producer. Choi et al. [18] provided an overview of threats of Interest flooding attacks for non-existent contents on NDN. Authors simulated and explained the effect of Interest flooding DoS attacks by a simple scenario over the quality of services for legitimate Interest packets from normal users due to PIT full. However, they do not analyze DoS attacks and their countermeasures.

5. RBF neural networks

Radial Basis Function (RBF) is a kind of feed-forward neural networks, which were developed by Broomhead and Lowe in 1998 [64]. This type of neural networks use a supervised algorithm and have been broadly employed for classification and interpolation regression [65]. As compared to other neural networks, RBF neural networks have better approximation characteristics, faster training procedures and simple network architecture. For these reasons, researchers have continued working on improving the performance of RBF learning algorithms [66, 67]. The RBF neural networks have three layers architecture including a single hidden layer of units. The first layer has n input units which connects the input space to the hidden layer. The hidden layer has m RBF units, which transforms the input units to the output layer. The output layer, consisting of l linear units. The output layer implements a weighted sum of hidden unit outputs. The input layer is non-linear while the output is linear. Due to non-linear approximation properties in RBF, this type of networks are able to model the complex mappings [34, 37]. The real output in output layer is given by:

$$y_s(X) = \sum_{j=1}^{k} w_{js}\phi\left(\frac{\left\|P - C_j\right\|}{\sigma_j}\right) \quad for\ 1 \le s \le l \qquad (1)$$

Where y_s is s-th network output, P is an input pattern, w_{js} is the weight of the link between j-th hidden neuron and s-th output neuron, C_j is the center of the j-th RBF unit in the hidden layer, and σ_j is the width of the j-th unit in the hidden layer. The ϕ denotes to an basis

(activation) function. The Gaussian activation function is used in this paper, which is given by [68]:

$$\phi_j(r) = exp(-\frac{\left\|P - C_j\right\|^2}{2\sigma_j^2}) \quad j = 1, 2, 3, ..., p \qquad (2)$$

Where r is the variable of radial basis function (ϕ).

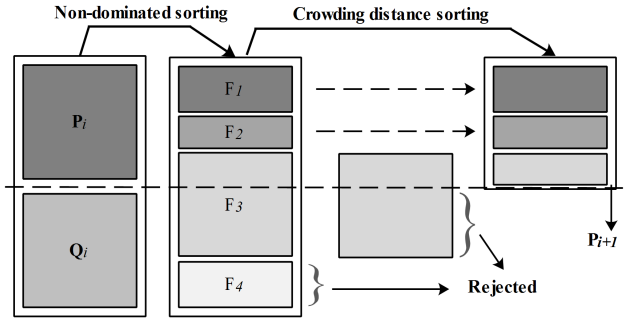


Figure 2: Schematic of the NSGA II procedure

6. Particle Swarm Optimization (PSO)

PSO was initially introduced in 1995 by James Kennedy and Russell Eberhart as a global optimization technique [69]. It was inspired by the social behavior of a bird flock or fish school. It is a population based meta-heuristic method that optimizes a problem by initializing a flock of birds randomly over the search space where each bird is referred as a *particle* and the population of particles is called *swarm* [70]. Each particle is a candidate solution to the problem which is assigned a velocity vector and has a memory which helps it in remembering its previous best position (known as local best, *lbest*). The particles move iteratively around in the search space according to a simple mathematical formula over the particle's position and velocity to find the global best position (known as *Gbest*). In the n-dimensional search space, the position and the velocity of i-th particle at t-th iteration of algorithm is denoted by vector $X_i(t) = (x_{i1}(t), x_{i2}(t), ..., x_{in}(t))$ and vector $V_i(t) = (v_{i1}(t), v_{i2}(t), ..., v_{in}(t))$, respectively. Usually a fitness function is the objective function to be minimized or maximized. Hereafter, a record of the best position of particle based on the fitness function value is kept in process. The best previously visited position of the particle i at current stage is denoted by vector $lbest_i = (lbest_{i1}, lbest_{i2}, ..., lbest_{in})$ as the personal best. The position of the best objective function value until the current stage is also recorded as the global best position denoted by $Gbest = (gbest_1, gbest_2, ..., gbest_n)$. The velocity and position of particle i at iteration $k+1$ can be calculated according the following equations:

$$\begin{aligned} V_i(k+1) = \omega V_i(k) + c_1 r_1 (lbest_i(k) - X_i(k)) \\ + c_2 r_2 (gbest(k) - X_i(k)) \end{aligned} \qquad (3)$$

$$X_i(k+1) = X_i(k) + V_i(k+1) \qquad (4)$$

Where ω is the inertia weight, $c1$ (cognitive parameter) and $c2$ (social parameter) are constants which control the search space between the local best position and the global best position (generally $c1 = c2 = 2$ [48]). Parameters r_1 and r_2 are random numbers uniformly distributed within [0 1]. Since larger ω performs more efficient global search and smaller one performs more effective local search, Eberhart and Shi [71] used Eq. (5) that properly balances between exploration (global search) and exploitation (local search) to avoid premature convergence to a local optimum.

$$\omega = \omega_{max} - t \cdot \frac{(\omega_{max} - \omega_{min})}{T} \qquad (5)$$

Where $\omega_{max}, \omega_{min}, T$ and t denote the maximum inertia weight, the minimum inertia weight, the total and the current number of iterations, respectively.

7. Non-dominated Sorting Genetic Algorithm (NSGA) II

NSGA II is one of the most widely and popular multi-objective optimization algorithms with three considerable properties including fast non-dominated sorting approach, fast crowded distance estimation procedure and simple crowded comparison operator [72]. Fig. 2 shows the NSGA II procedure. Generally, NSGA II can be roughly detailed as following steps [72, 73, 74]:

Step 1: Population initialization

A set of random solutions (chromosomes) with a uniform distribution based on the problem range and constraint are generated. The first generation is a $N \times D$ matrix. N and D are identified as the number of chromosomes and decision variables (genes), respectively.

Step 2: Non-dominated sort

Sorting process based on non domination criteria of the initialized population.

Step 3: Crowding distance

Chromosomes are classified to the Pareto fronts using:

$$d_{I_j} = \sum_{m=1}^{M} \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{Max} - f_m^{Min}} \qquad (6)$$

Where, d_{I_j} is crowded distance of jth solution, M is number of objectives, $f_m^{I_{j+1}^m}$ and $f_m^{I_{j-1}^m}$ are values of mth

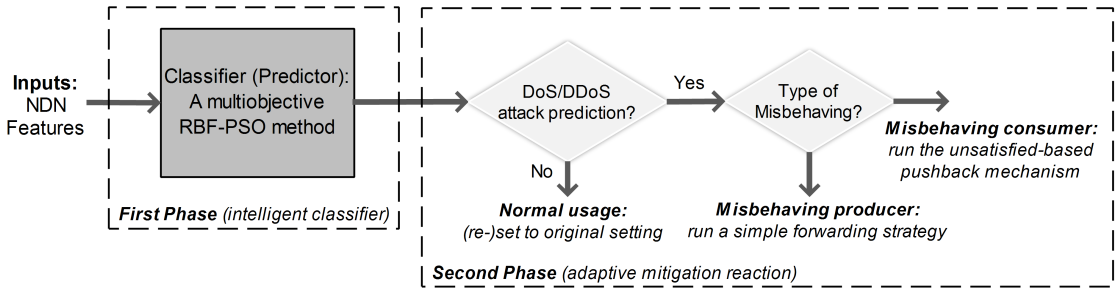


Figure 3: The overview of the proposed DoS mitigation method in NDN

objective for $(j-1)$th and $(j+1)$th solution, f_m^{Max} is maximum value of mth objective function among solutions of the current population, f_m^{Min} is minimum value of mth objective function among solutions of the current population, I_j is the jth solution in the sorted list and $(j-1)$ and $(j+1)$ are two nearest neighboring solutions on both sides of I_j. Afterwards, the algorithm searches the nearest points (solutions) with more value of d_{I_j}. Solutions in the best-known Pareto set should be uniformly distributed and diverse over of the Pareto front in order to provide the decision maker a true picture of trade-offs. Then, Pareto fronts are ranked from the best to the worst.

Step 4: Selection

The selection of chromosomes is carried out to select appropriate chromosomes (parents) using the crowded tournament operator. The crowded tournament operator compares different solutions with two criteria, (1) a non-dominated rank and (2) a crowding distance in the population. In this process, if a solution dominates the others, it will be selected as the parent. Otherwise, the solution with the higher value of crowding distance (highest diversity) will be selected.

Step 5: Genetic algorithm operators

There are a variety of recombination (crossover) and mutation operators.

Step 6: Recombination and selection

The offspring population is combined with the current generation population and the total population is sorted based on non-domination. The new generation is filled by chromosomes from each front subsequently until the population size exceeds the current population size N.

8. The proposed hybrid intelligent method for DoS mitigation in NDN

In this section, we introduce our method, a two-phase framework for mitigating DoS attacks in NDN. The first phase being proactive detection (see section 8.1) and the second one adaptive reaction (see section 11.2). The proposed predictor in the first phase is a global framework so that we can use the predictor in other networks. In this paper, we apply the proposed predictor successfully on some benchmark problems and NDN and leave further investigations in other networks to future work. A diagram of the two phases of the proposed method is shown in Fig. 3.

8.1. The proposed intelligent classifier (predictor)

This section presents the details of proposed intelligent algorithm for classification problems. Our approach composes of two main phases. It is depicted in Fig. 4. Each phase is given in the next subsections.

8.1.1. Phase 1: Improvement of RBF parameters

In the first phase -training (optimization)- we introduce a new hybrid optimization approach for designing RBF neural networks which can be implemented for real-world problems. Firstly, a new multiobjective optimization algorithm as NSGA II for adjusting centers of the RBF units is introduced. This algorithm obtains various non-dominated sets that provide an appropriate balance between two conflicting objectives: well-separated and local optimization of RBF centers. Secondly, PSO algorithm has been applied to simultaneously tune widths of the RBF units and output weights through well-placed centers. The algorithm is presented below:

A. First part (adjusting RBF units' centers based on NSGA II):

1. Problem definition:
   1-1- population size (N), maximum iteration ($Iter_{Max}$), crossover percentage ($pCrossover$), number of parents (offspring) after crossover operator ($nCrossover = 2 \times round(pCrossover \times \frac{N}{2})$), mutation percentage ($pMutation$), number of mutants after mutation ($nMutation =$

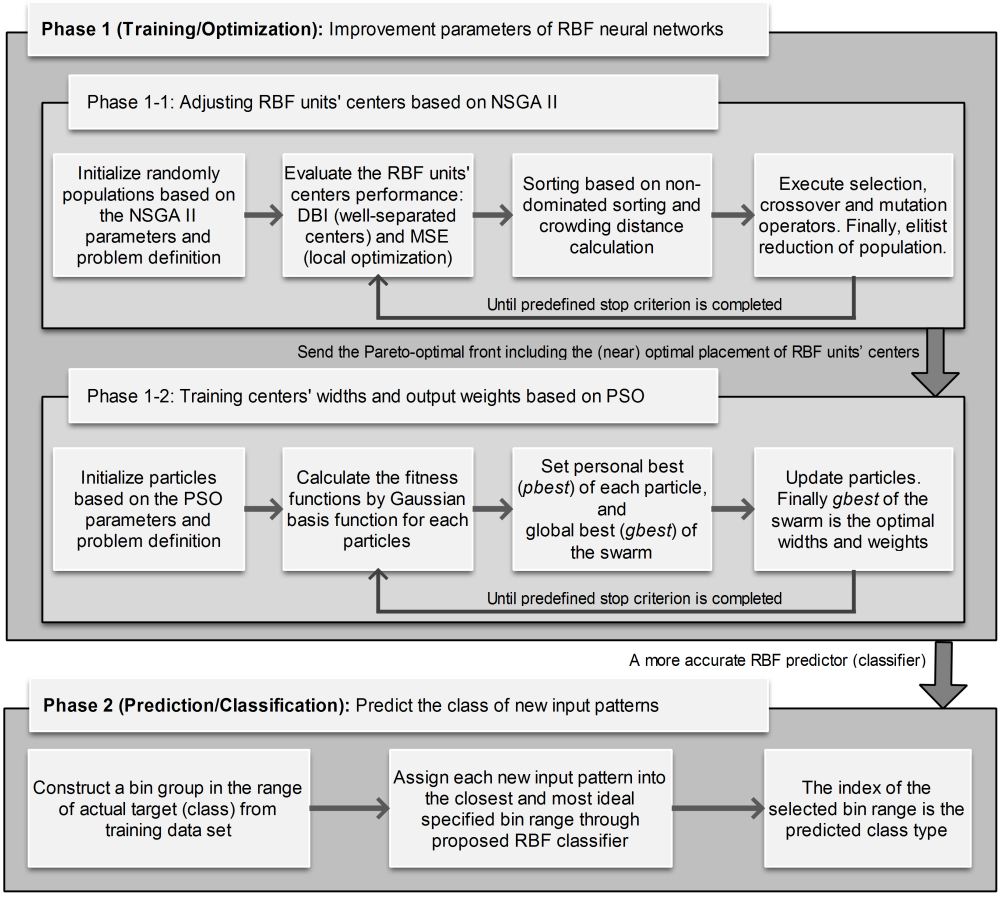


Figure 4: Proposed intelligent algorithm for more accurate classification

$round(pMutation \times N)$), mutation rate ($mu$), mutation step size ($sigma = 0.1$).

2. Initialize population:
   2-1- Generate the initial populations (individuals) P, including $P_1, P_2, ..., P_N$.
   2-2- Calculate the two conflicting cost functions as DBI and MSE (presented in section 8.1.3) for each population.
   2-3- Rank all populations according to their non-dominance.
   2-4- Calculate the crowding distances for all populations to keep the population diversity (Eq. 6).
   2-5- Sort the non-dominated solutions in descending crowding distance and rank values.
3. NSGA II main loop:
   3-1- Execute the evolution process including crossover and mutation operators:
   a. Execute crossover operator, $PopCrossover$ (this paper adopts the two-point crossover).
   b. Execute mutation operator, $PopMutation$ (A Gaussian distributed random number with mean zero and variance 1 is used [75, 76]).
   c. Merge populations:
   $P = [P \;\; PopCrossover \;\; PopMutation]$.
   3-2- Run steps 2-3 (rank), 2-4 (crowding distance) and 2-5 (sort) over the merged P.
   3-3- Truncate/Select the generated population P to the range of population size: $P = P(1 : N)$.
   3-4- Run steps 2-3 (rank), 2-4 (crowding distance) and 2-5 (sort) over the truncated P.
   3-5- Store Pareto-optimal front (non-dominated set) in the archive as $PF1$.
   3-6- Repeat Step 3 until termination condition ($Iter_{Max}$) is reached.
   3-7- Keep the final $PF1$ including the (near) optimal placement of RBF units' centers.

B. Second part (calculating widths of the RBF units and output weights based on PSO algorithm):

1. Problem definition:

1-1- population size (N), maximum iteration ($Iter_{Max}$) and number of RBF Kernel obtained from $PF1$ in phase A ($nKernel$).
1-2- Upper and lower bound of width (σ) and weight (w) variables.
1-3- Adjust the PSO parameters: inertia weight (ω) which is linearly decrease by Eq. 5, acceleration coefficients ($c1 = c2 = 2$), and two random numbers ($r1$ and $r2$) which distributed uniformly in [0 1].

2. Initialize population for each particle:
2-1- Generate the initial populations (particle positions) including $Particle(1), Particle(2), ..., Particle(N)$:
$Particle(i)$.Position.σ and $Particle(i)$.Position.w. $i = 1, 2, ..., N$.
$particle(i)$.Position.σ = Continuous uniform random numbers between $\sigma.Lower$ and $\sigma.Upper$ in size of $nKernel$.
$particle(i)$.Position.w = Continuous uniform random numbers between $w.Lower$ and $w.Upper$ in size of $nKernel$.
2-2- Initialize velocity vectors in a feasible space for each particle:
$particle(i)$.Velocity.σ = a $nKernal$ size zero matrix.
$particle(i)$.Velocity.w = a $nKernal$ size zero matrix.
2-3- Evaluate each particle by Gaussian basis function in each RBF units (Eq. 1). Calculate Gaussian basis function with two tuned parameters (σ -centers' widths- and w -output weights- from PSO) and optimal placement of RBF units' centers from archive $PF1$.
2-4- Initially, personal best ($lbest$) is the current calculated cost.

3. Set the global best ($gbest$) to a particle with the lowest cost.

4. PSO main loop:
4-1- Update velocity for each particle by Eq. 3.
4-2- Control the lower (V_{min}) and upper (V_{max}) bounds of velocity:
$V_{min} \leq V_i t \leq V_{max}$. Where, i (particle id)=1, 2, ..., N and t (iteration number)=1, 2, ..., $Iter_{Max}$.
4-3- Update position by Eq. 4.
4-4- If the current velocity and position are outside of the boundaries, they take the upper bound or lower bound. They are multiplied by -1 so that they search in the opposite direction (mirroring to feasible search space).
4-5- Update personal best ($lbest$): if the current particle cost is better than the previous (recorded in $lbest$) particle cost, then set the current particle cost as the personal best.
4-6- Update global best ($gbest$): if the current personal best is better than the global best, then set the current personal best as the global best in the swarm.

5. Repeat Step 4 until termination condition (Max_{Iter}) is reached. Otherwise, $gbest$ is the optimized RBF units' widths and output weights.

8.1.2. Phase 2: classification of new input patterns

In the second phase -prediction (classification)- we classify (predict) the class type of new input patterns, which we do not know about their target classes in prior. The classification is calculated by defining bins. Data samples should be normalized into [0 1], when dealing with parameters of different units and scales [77]. Since data set is normalized in range of [0 1], bin values should be defined in this range. The number of bin ranges are equal to the number of target classes in training phase. Then, we can determine which data object falls into a specified bin range. For instance, if the number of target class in a particular data set is five classes, then the range of bin values can be organized in the range of [0 0.25 0.5 0.75 1]. Hereafter, constructed RBF neural network from first phase is executed over the input patterns. The RBF output is always a decimal number between [0 1]. This output assigns to the closest and most ideal index of specified bin range, e.g., if output=0.65, then the input pattern falls into fourth bin. It means that the predicted class is four. The pseudo-code of classification computation is given below:

1- Define some input parameters:
$LowEdge$ = lower bound of target class.
$UpEdge$ = upper bound of target class.
$NumBins$= number of target classes in training data set.
$BinEdges$= Generate linearly spaced vectors between $LowEdge$ and $UpEdge$ in the size of $NumBins$, where the bin range is equal to the number of target class.
2- Assign input patterns into the closest index of specified bin range. The index of bin range is the predicted classes of input patterns.

Table 1: The four applied benchmark data sets

Data set	No. of features	No. of classes	No. of patterns
Wine	13	3	178
Iris	4	3	150
Ionosphere	34	2	351
Zoo	17	7	101

8.1.3. Objective functions in NSGA II

Two objective functions are used to evaluate the RBF network units' centers performance. The two objective functions for minimization problems are:

1. Local optimization based on Mean Square Error (MSE):

Given the set of centers (c), the set of corresponding data objects (x), c_x denotes the center corresponding to the x, and N is the number of data points, MSE can be calculated as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} d(x_i, c_x)^2 \tag{7}$$

2. Well-separated (well-placed) RBF units' centers based on Davies-Boulding Index (DBI).

Based on our experiments [46], we have found it quite reliable. DBI [78] takes into account both compactness and separation criteria that makes similar data points within the same centers and places other data points in distinct centers. The compactness of a group of data objects with corresponding center is calculated based on the MSE. The separation is measured by the distance between centers c_i and c_j. In general, the DBI is given by:

$$\frac{1}{NC}\sum_{i} max_{j,j\neq i}\frac{[\frac{1}{n_i}\sum_{x\epsilon C_i} d(x, c_i) + \frac{1}{n_j}\sum_{x\epsilon C_j} d(x, c_j)]}{d(c_i, c_j)} \tag{8}$$

Where, NC is the number of centers, x is the corresponding data objects, n_i is the number of data objects belonging to the center c_i.

Table 2: adjusting RBF units' centers in Wine

n	Pop.	Iter.	MSE	Std.	SEM	CI (95%)
PSO:						
20	20	1500	0.19224	0.1235	0.0101	[0.182 0.671]
40	30	2000	0.16474	0.1207	0.0104	[0.176 0.649]
70	35	2500	0.14989	0.1013	0.0088	[0.165 0.572]
GA:						
20	20	1500	0.19423	0.1242	0.0104	[0.188 0.68]
40	30	2000	0.16532	0.1222	0.0105	[0.196 0.671]
70	35	2500	0.3729	0.1072	0.0093	[0.171 0.583]
ICA:						
20	20	1500	0.41448	0.1421	0.0123	[0.349 0.907]
40	30	2000	0.3396	0.1235	0.0107	[0.327 0.812]
70	35	2500	0.30012	0.124	0.0107	[0.291 0.777]
DE:						
20	20	1500	0.38732	0.1484	0.0128	[0.314 0.895]
40	30	2000	0.41173	0.1555	0.0134	[0.318 0.928]
70	35	2500	0.41586	0.1442	0.0125	[0.346 0.911]

9. Benchmarking the proposed intelligent classifier (predictor)

For assurance of robustness and accuracy of our proposed intelligent hybrid classifier (predictor), we applied the four classic benchmark problems from the UCI machine learning repository [79]. Table 1 shows the main characteristics of these data sets. In the experiments, 70% of data set is used as training data set and the rest is considered as testing data set in order to validate the functionality of the proposed method. We evaluated different performance criteria including Mean Square Error (MSE), Standard Deviation (Std.),

Table 3: adjusting RBF units' centers in Iris

n	Pop.	Iter.	MSE	Std.	SEM	CI (95%)
PSO:						
25	35	2000	0.00957	0.0309	0.0036	[0.009 0.169]
35	50	2500	0.00661	0.0346	0.0033	[0.006 0.142]
40	70	3000	0.00541	0.0362	0.0032	[0.007 0.135]
GA:						
25	35	2000	0.01078	0.0394	0.0037	[0.019 0.173]
35	50	2500	0.00975	0.0439	0.0041	[0.0072 0.175]
40	70	3000	0.00598	0.0365	0.0033	[-0.001 0.138]
ICA:						
25	35	2000	0.02359	0.0666	0.0063	[0.011 0.269]
35	50	2500	0.01497	0.0438	0.0041	[-0.209 0.239]
40	70	3000	0.01332	0.0368	0.0035	[0.037 0.182]
DE:						
25	35	2000	0.02396	0.0595	0.0056	[0.026 0.26]
35	50	2500	0.02376	0.049	0.0046	[0.05 0.242]
40	70	3000	0.02352	0.0584	0.0055	[0.131 0.153]

Table 4: adjusting RBF units' centers in Ionosphere

n	Pop.	Iter.	MSE	Std.	SEM	CI (95%)
PSO:						
40	60	3000	0.90357	0.4709	0.0297	[-0.104 1.763]
50	80	4000	0.81119	0.457	0.0282	[-0.079 1.673]
60	90	4000	0.74164	0.4496	0.0284	[-0.085 1.631]
GA:						
40	60	3000	1.043	0.4953	0.0299	[-0.111 1.836]
50	80	4000	0.9489	0.4615	0.0285	[-0.086 1.763]
60	90	4000	0.9394	0.4501	0.0278	[-0.093 1.741]
ICA:						
40	60	3000	2.113	0.5575	0.0344	[0.25 2.436]
50	80	4000	1.9462	0.4792	0.0295	[0.37 2.25]
60	90	4000	1.8535	0.4743	0.0292	[0.347 2.206]
DE:						
40	60	3000	2.6211	0.5671	0.035	[0.405 2.629]
50	80	4000	2.6249	0.5878	0.0362	[0.358 2.663]
60	90	4000	2.5915	0.5493	0.0339	[0.437 2.59]

Table 5: adjusting RBF units' centers in Zoo

n	Pop.	Iter.	MSE	Std.	SEM	CI (95%)
PSO:						
40	50	2000	0.75405	0.23	0.0264	[-0.288 1.289]
50	70	2500	0.67409	0.2622	0.0301	[0.198 1.318]
60	90	3000	0.68884	0.2563	0.0274	[0.249 1.253]
GA:						
40	50	2000	0.75469	0.2793	0.032	[0.296 1.371]
50	70	2500	0.68008	0.3057	0.0351	[0.201 1.366]
60	90	3000	0.69329	0.2654	0.0281	[0.315 1.277]
ICA:						
40	50	2000	1.1539	0.303	0.0348	[0.315 1.377]
50	70	2500	0.9867	0.3112	0.0357	[0.334 1.554]
60	90	3000	1.0088	0.2826	0.0324	[0.41 1.518]
DE:						
40	50	2000	1.96	0.3213	0.0369	[0.733 1.933]
50	70	2500	1.9406	0.2829	0.0325	[0.81 1.919]
60	90	3000	1.8115	0.2736	0.0314	[0.782 1.855]

Table 6: Classification of Wine data set based on RBF-PSO optimization algorithm

n	Pop.	Iter.	Training data set					Test data set				
			MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
Units' centers by PSO:												
20	25	2000	0.00838	0.0912	[-0.157 0.158]	0.00692	2	0.01078	0.109	[-0.208 0.2]	0.01567	2
40	30	2500	0.00586	0.08024	[-0.158 0.157]	0.00676	0	0.01389	0.11475	[-0.25 0.199]	0.01814	3
70	40	3000	0.00519	0.07145	[-0.135 0.146]	0.00617	1	0.01316	0.11656	[-0.22 0.237]	0.0174	3
Units' centers by GA:												
20	25	2000	0.0084	0.093	[-0.164 0.166]	0.00725	1	0.01082	0.10907	[-0.216 0.192]	0.01568	3
40	30	2500	0.00598	0.08227	[-0.171 0.173]	0.00624	1	0.01479	0.11874	[-0.265 0.201]	0.0179	4
70	40	3000	0.00525	0.07254	[-0.137 0.148]	0.00626	2	0.01501	0.12183	[-0.261 0.216]	0.01836	3
Units' centers by ICA:												
20	25	2000	0.00917	0.09615	[-0.188 0.189]	0.0083	1	0.01688	0.13139	[-0.262 0.254]	0.0198	4
40	30	2500	0.00716	0.08496	[-0.166 0.167]	0.00734	1	0.01483	0.11884	[-0.234 0.216]	0.01742	3
70	40	3000	0.00677	0.08255	[-0.159 0.165]	0.00713	1	0.01576	0.12683	[-0.038 0.024]	0.01912	3
Units' centers by DE:												
20	25	2000	0.01159	0.10808	[-0.212 0.212]	0.00933	2	0.02135	0.14608	[-0.309 0.264]	0.02202	3
40	30	2500	0.00906	0.09555	[-0.187 0.188]	0.00825	1	0.01401	0.11895	[-0.252 0.211]	0.01778	3
70	40	3000	0.00648	0.08082	[-0.159 0.158]	0.00698	2	0.01327	0.11926	[-0.256 0.211]	0.01787	3

Table 7: Classification of Iris data set based on RBF-PSO optimization algorithm

n	Pop.	Iter.	Training data set					Test data set				
			MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
Units' centers by PSO:												
25	35	2000	0.007	0.07407	[-0.175 0.125]	0.0079	2	0.01347	0.10954	[0.211 0.219]	0.01844	3
35	50	2500	0.00435	0.06626	[-0.13 0.13]	0.00623	2	0.01419	0.09469	[-0.189 0.182]	0.01692	2
40	70	3000	0.00429	0.07007	[-0.132 0.131]	0.00682	3	0.05785	0.08827	[-0.189 0.157]	0.01551	2
Units' centers by GA:												
25	35	2000	0.00781	0.08877	[-0.174 0.174]	0.00835	2	0.01406	0.11579	[-0.223 0.231]	0.01903	5
35	50	2500	0.00454	0.06768	[-0.132 0.133]	0.00636	1	0.01416	0.10704	[-0.206 0.213]	0.01759	3
40	70	3000	0.00432	0.07391	[-0.145 0.145]	0.00544	2	0.05557	0.0734	[-0.162 0.126]	0.01206	2
Units' centers by ICA:												
25	35	2000	0.0079	0.08717	[-0.153 0.149]	0.00725	2	0.01517	0.12285	[-0.238 0.243]	0.01897	3
35	50	2500	0.00543	0.07405	[-0.146 0.145]	0.00696	3	0.01542	0.12586	[-0.244 0.249]	0.02069	2
40	70	3000	0.00455	0.07008	[-0.137 0.137]	0.00563	2	0.05092	0.10541	[-0.217 0.196]	0.01732	3
Units' centers by DE:												
25	35	2000	0.00782	0.08188	[-0.149 0.149]	0.00721	2	0.01378	0.11638	[-0.15 0.15]	0.01913	3
35	50	2500	0.00493	0.07666	[-0.123 0.124]	0.00592	2	0.01822	0.09973	[-0.202 0.189]	0.01608	2
40	70	3000	0.00625	0.07941	[-0.153 0.158]	0.00747	3	0.05956	0.09912	[-0.192 0.197]	0.01629	3

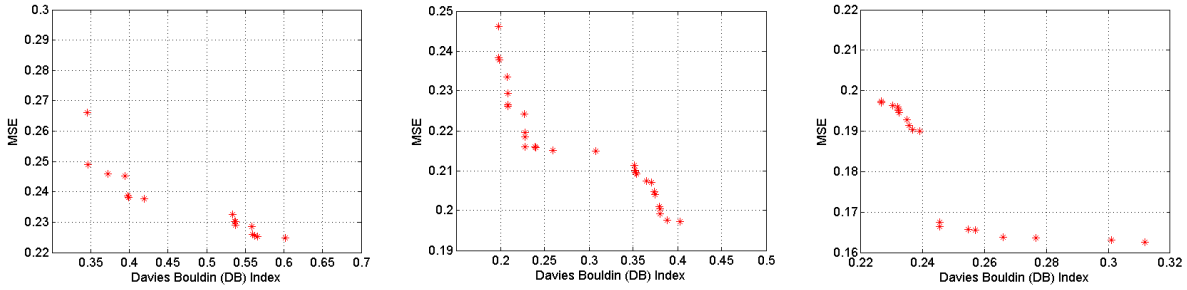
Table 8: Classification of Ionosphere data set based on RBF-PSO optimization algorithm

n	Pop.	Iter.	Training data set					Test data set				
			MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
Units' centers by PSO:												
30	60	2000	0.05403	0.22663	[-0.444 0.444]	0.01395	17	0.05838	0.2387	[-0.423 0.513]	0.02544	5
40	80	2500	0.05172	0.22785	[-0.448 0.445]	0.01405	16	0.05553	0.23233	[-0.409 0.502]	0.024767	3
50	90	3000	0.0466	0.20488	[-0.401 0.403]	0.01244	14	0.04855	0.21235	[-0.373 0.459]	0.02289	4
Units' centers by GA:												
30	60	2000	0.05464	0.23407	[-0.451 0.467]	0.01443	19	0.05436	0.22923	[-0.395 0.504]	0.02443	5
40	80	2500	0.06114	0.24773	[-0.485 0.487]	0.01527	20	0.07284	0.27006	[-0.502 0.557]	0.02878	7
50	90	3000	0.05673	0.23859	[-0.473 0.462]	0.01471	17	0.05943	0.24339	[-0.448 0.507]	0.02594	4
Units' centers by ICA:												
30	60	2000	0.07042	0.2658	[-0.528 0.514]	0.01639	19	0.06913	0.26367	[-0.497 0.537]	0.0281	7
40	80	2500	0.06699	0.25932	[-0.508 0.509]	0.01599	20	0.06238	0.24981	[-0.427 0.552]	0.02662	5
50	90	3000	0.06389	0.25318	[-0.491 0.502]	0.01561	21	0.06058	0.24449	[-0.441 0.518]	0.026	5
Units' centers by DE:												
30	60	2000	0.05847	0.24228	[-0.474 0.476]	0.01492	18	0.06325	0.24815	[-0.438 0.535]	0.02645	4
40	80	2500	0.05798	0.24125	[-0.474 0.472]	0.01487	15	0.05386	0.22932	[-0.406 0.493]	0.02444	3
50	90	3000	0.06175	0.24898	[-0.489 0.487]	0.01535	16	0.06379	0.2508	[-0.452 0.532]	0.02673	6

Standard Error of Mean (SEM), Confidence Interval (CI) by 95% and the number of incorrect classification (Cls. err.). Firstly, we adjust RBF units' centers based on MSE as a frequently used cost function (minimization objective) in the literature. We employ four optimization algorithms which are widely used and adopted successfully in different applications including PSO [80, 81, 82, 83, 84, 85], Genetic Algorithm (GA) [86, 87, 88, 89, 90, 91], Imperialist Competitive Algorithm (ICA) [92, 93, 94] and Differential Evolution (DE) [95, 96, 97, 98]. The experiments on each algorithm were repeated 20 times independently to find

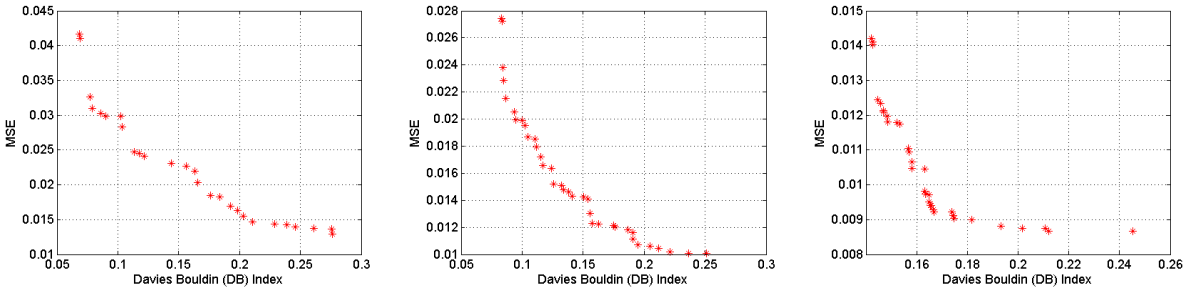
Table 9: Classification of Zoo data set based on RBF-PSO optimization algorithm

n	Pop.	Iter.	Training data set					Test data set				
			MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
Units' centers by PSO:												
30	50	2000	0.00156	0.03974	[-0.077 0.079]	0.00455	3	0.00394	0.0552	[-0.113 0.104]	0.01104	3
40	70	2500	0.00093	0.03024	[-0.059 0.059]	0.00366	1	0.00471	0.07	[-0.14 0.135]	0.01401	5
50	90	3000	0.00095	0.03114	[-0.061 0.061]	0.00335	3	0.00607	0.07197	[-0.157 0.131]	0.01439	4
Units' centers by GA:												
30	50	2000	0.00222	0.47477	[-0.093 0.094]	0.00544	4	0.00904	0.08952	[-0.212 0.139]	0.0179	7
40	70	2500	0.00141	0.03783	[-0.074 0.074]	0.00433	5	0.00595	0.07695	[-0.167 0.134]	0.01539	4
50	90	3000	0.00115	0.03422	[-0.067 0.067]	0.00392	4	0.00628	0.06858	[-0.147 0.122]	0.01383	4
Units' centers by ICA:												
30	50	2000	0.00211	0.04628	[-0.089 0.093]	0.0053	5	0.00706	0.08286	[-0.155 0.17]	0.01657	4
40	70	2500	0.0011	0.0334	[-0.065 0.066]	0.00383	2	0.00487	0.06209	[-0.155 0.17]	0.01241	6
50	90	3000	0.00102	0.03228	[-0.063 0.064]	0.0037	2	0.00826	0.08935	[-0.2 0.151]	0.01787	4
Units' centers by DE:												
30	50	2000	0.00159	0.04024	[-0.079 0.078]	0.00461	3	0.00514	0.07046	[-0.158 0.119]	0.01409	5
40	70	2500	0.00136	0.03712	[-0.073 0.073]	0.00425	4	0.00848	0.08664	[-0.206 0.134]	0.01733	6
50	90	3000	00113	0.03385	[-0.066 0.066]	0.00388	2	0.00635	0.074	[-0.155 0.135]	0.0148	4



(a) Multi objective (n=20) (b) Multi objective (n=40) (c) Multi objective (n=70)

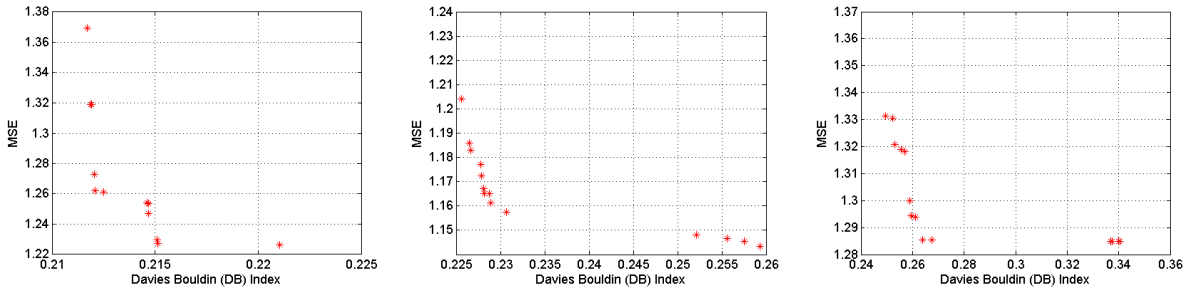
Figure 5: Optimal Pareto fronts of Wine data set



(a) Multi objective (n=20) (b) Multi objective (n=50) (c) Multi objective (n=70)

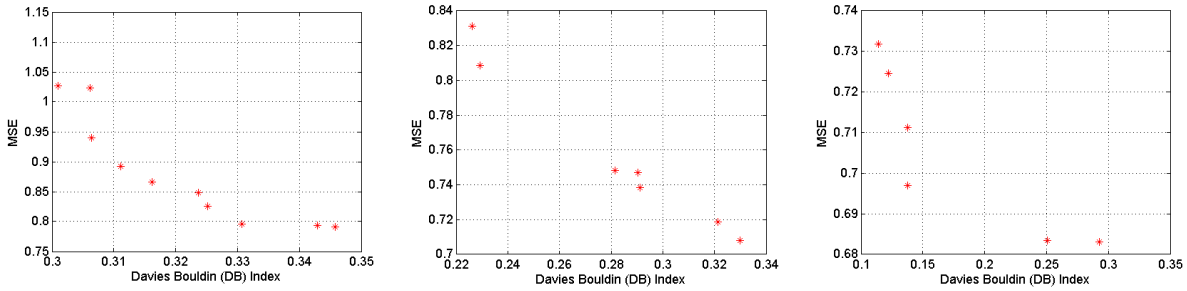
Figure 6: Optimal Pareto fronts of Iris data set

the optimal considered performance criteria. Tables 2-5 show the comparison of (best) results over applied benchmarking problems. As seen in these Tables, PSO performs better results in estimation of RBF units' centers as compared to others based on the applied performance measures. The second optimal results have also performed by GA. However, we have evaluated all results as the (near) optimal adjustment of units' centers for adjusting two others RBF network parameters. Secondly, for adjusting the RBF units' widths and output weights, we integrate the optimal placement of centers from four applied optimization algorithm (from Tables 2-5) with PSO. The obtained results are shown in Tables 6-9. The classification error (Cls. err.) is calculated based on our proposed algorithm in the second phase. As seen in these tables, PSO is almost able to



(a) Multi objective (n=60) (b) Multi objective (n=80) (c) Multi objective (n=90)

Figure 7: Optimal Pareto fronts of Ionosphere data set



(a) Multi objective (n=50) (b) Multi objective (n=70) (c) Multi objective (n=90)

Figure 8: Optimal Pareto fronts of Zoo data set

achieve better results than the other methods in terms of the classification error and other applied metrics. Experimental results demonstrate that even though the ICA and the DE with not so proper results in obtaining RBF units' centers could successfully provide low classification error. Unlike the suitable number of correct classification by ICA and DE, they do not usually perform well in terms of MSE, Std., CI (95%) and SEM as compared to PSO and GA. Since the number of correct classification is the major criterion in the classification problems, it can be concluded that the MSE (as minimization objective) is not a suitable performance metric for finding the (near) optimal placement of units' centers. To confirm convincingly this claim, this paper presents a multi-objective approach to find the (near) optimal placement of centers. According to the first part of the proposed method (see Fig. 4), NSGA II was applied over benchmarking problems by two conflicting objectives (DBI and MSE) in order to find the well-separated centers and their local optimization, respectively. The experiment on proposed algorithm was repeated 5 times independently to find the optimal performance metrics. Figs. 5-8 are depicted the optimal Pareto front solutions of (near) well-placed of RBF units' centers through DBI (x-axis) and MSE (y-axis). We are going to show that for constructing final RBF neural networks, MSE is not solely the ideal accurate criterion.

Afterward, we integrate the optimal placement of units' centers (obtained by our two-objective approach in Figs. 5-8) with the PSO (see second step of the first phase in Fig. 4) in order to optimize and tune units' widths and output weights. We run the PSO algorithm with all the optimal Pareto front solutions of units' centers. The first five optimal results are demonstrated based on the minimum classification error in both training and testing data sets in Tables 10-13. As seen in these tables, the first five optimal Pareto solutions outperform significantly the other methods by single-objective approach in Tables 6-9 based on the MSE, Std. and the number of misclassification error. Other applied performance metrics outperforms the single-objective approach over 90% of results. The results show that the proposed method can provide several well-placed RBF units' centers as compared to the traditional (single-objective) approaches through MSE criterion. To sum up, MSE is not an unique criterion to evaluate the performance of

Table 10: Classification of Wine data set based on proposed method

n	Pop.	Iter.	NSGA II		Training data set					Test data set				
			MSE	DBI	MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
			0.2252	0.566	0.00637	0.08015	[-0.157 0.158]	0.00692	0	0.01042	0.10331	[-0.222 0.191]	0.01708	1
			0.2249	0.6022	0.00702	0.08113	[-0.164 0.166]	0.00726	0	0.01051	0.10156	[-0.227 0.2]	0.01832	1
20	25	2500	0.2303	0.5362	0.00734	0.08134	[-0.169 0.17]	0.00745	0	0.01055	0.10433	[-0.232 0.201]	0.01874	1
			0.2376	0.4196	0.00768	0.08196	[-0.162 0.171]	0.00759	1	0.01062	0.10871	[-0.23 0.21]	0.01804	1
			0.2452	0.3941	0.00814	0.08357	[-0.167 0.174]	0.00782	1	0.01036	0.10769	[-0.217 0.202]	0.01623	2
			0.2004	0.3799	0.0049	0.07027	[-0.138 0.138]	0.00607	0	0.01074	0.10322	[-0.221 0.184]	0.01556	1
			0.1973	0.4032	0.00491	0.07031	[-0.137 0.139]	0.00607	0	0.01312	0.11353	[-0.246 0.2]	0.01631	1
40	30	2500	0.1993	0.3803	0.0051	0.07657	[-0.149 0.151]	0.00661	0	0.01382	0.11436	[-0.247 0.201]	0.01661	1
			0.2095	0.3532	0.00527	0.07952	[-0.155 0.157]	0.00687	0	0.01316	0.11438	[-0.251 0.198]	0.01669	1
			0.2074	0.3646	0.00556	0.07981	[-0.156 0.157]	0.00702	0	0.01385	0.11311	[-0.259 0.185]	0.01625	2
			0.1638	0.2661	0.00397	0.06326	[-0.125 0.123]	0.00546	0	0.01262	0.11062	[-0.243 0.191]	0.01667	1
			0.1657	0.2547	0.00442	0.06675	[-0.131 0.131]	0.00576	0	0.01263	0.11006	[-0.244 0.188]	0.01659	1
70	40	3500	0.1636	0.2767	0.00445	0.06692	[-0.128 0.135]	0.00578	0	0.01298	0.11387	[-0.241 0.205]	0.01716	1
			0.1630	0.3011	0.00456	0.06781	[-0.133 0.132]	0.00585	0	0.01276	0.11386	[-0.233 0.213]	0.01716	2
			0.1674	0.2454	0.00511	0.06963	[-0.142 0.131]	0.00618	0	0.01315	0.11569	[-0.243 0.21]	0.01744	2

Table 11: Classification of Iris data set based on proposed method

n	Pop.	Iter.	NSGA II		Training data set					Test data set				
			MSE	DBI	MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
			0.0136	0.2759	0.00465	0.06852	[-0.134 0.134]	0.00644	1	0.01183	0.10909	[-0.198 0.23]	0.01793	2
			0.0129	0.0403	0.00527	0.07292	[-0.142 0.144]	0.00686	2	0.01285	0.10272	[-0.168 0.235]	0.02175	2
25	35	2500	0.0183	0.184	0.0055	0.07452	[-0.145 0.147]	0.00701	1	0.01268	0.10028	[-0.173 0.22]	0.02106	2
			0.0142	0.2386	0.00585	0.07486	[-0.15 0.151]	0.00723	1	0.00996	0.1012	[-0.197 0.2]	0.01663	2
			0.0169	0.1924	0.00632	0.07488	[-0.156 0.157]	0.00751	2	0.01241	0.10918	[-0.191 0.237]	0.02148	2
			0.0111	0.1911	0.00316	0.05651	[-0.111 0.111]	0.00531	1	0.01397	0.09296	[-0.153 0.191]	0.0205	1
			0.0121	0.1752	0.00344	0.05893	[-0.115 0.116]	0.00554	1	0.01394	0.09134	[-0.146 0.212]	0.02005	1
35	50	3000	0.0106	0.2048	0.00346	0.05908	[-0.115 0.116]	0.00555	1	0.01362	0.09189	[-0.162 0.191]	0.02066	1
			0.0116	0.1910	0.00377	0.06169	[-0.121 0.121]	0.0058	2	0.01295	0.09245	[-0.164 0.198]	0.02006	1
			0.0105	0.2122	0.00381	0.06206	[-0.123 0.121]	0.00583	1	0.01241	0.09067	[-0.16 0.196]	0.02076	1
			0.0095	0.1650	0.00416	0.06636	[-0.129 0.131]	0.00624	1	0.03992	0.05205	[-0.058 0.046]	0.0265	1
			0.0088	0.1932	0.00418	0.06798	[-0.133 0.134]	0.00639	1	0.03904	0.05666	[-0.053 0.169]	0.02561	1
40	70	3500	0.0091	0.1744	0.0042	0.06814	[-0.133 0.135]	0.00641	1	0.03476	0.05989	[-0.074 0.161]	0.0245	1
			0.0087	0.2108	0.00422	0.06829	[-0.132 0.135]	0.00642	1	0.03235	0.05287	[-0.056 0.152]	0.02335	1
			0.0097	0.1636	0.00423	0.06903	[-0.134 0.136]	0.00649	2	0.03468	0.04836	[-0.045 0.144]	0.02425	1

the units' centers in RBF networks. A new hybrid optimization approach for well-separated centers (such as by DBI) and their local optimization (such as by MSE) in estimation of RBF units' centers would fit considerably the performance requirements.

10. Evaluation environment

We use simulations to quantify effects of DoS attacks and their countermeasures. In this work, we used the open-source ndnSIM [99] package, which implements NDN protocol stack for NS-3 network simulator (`http://www.nsnam.org/`), to run simulations

Table 12: Classification of Ionosphere data set based on proposed method

n	Pop.	Iter.	NSGA II		Training data set					Test data set				
			MSE	DBI	MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
			1.2472	0.2147	0.05284	0.22464	[-0.442 0.439]	0.01346	12	0.05129	0.2064	[-0.371 0.438]	0.02139	2
			1.2266	0.2210	0.05322	0.22545	[-0.442 0.442]	0.01351	11	0.0538	0.20913	[-0.372 0.448]	0.02162	2
30	60	2000	1.2539	0.2147	0.05334	0.22571	[-0.443 0.442]	0.01353	12	0.05392	0.20226	[-0.366 0.472]	0.020957	2
			1.2297	0.2151	0.05338	0.22171	[-0.431 0.438]	0.0139	12	0.05308	0.21136	[-0.389 0.439]	0.02079	2
			1.262	0.2121	0.05339	0.22189	[-0.437 0.433]	0.01391	11	0.05366	0.21105	[-0.377 0.451]	0.02076	2
			1.1434	0.2593	0.04309	0.20793	[-0.403 0.413]	0.01282	9	0.03759	0.19029	[-0.331 0.415]	0.02028	2
			1.1651	0.2282	0.0437	0.20946	[-0.409 0.412]	0.01291	10	0.04206	0.20113	[-0.349 0.44]	0.02144	1
40	80	2500	1.1481	0.2521	0.04391	0.20994	[-0.41 0.413]	0.01294	9	0.04378	0.20484	[-0.354 0.45]	0.02183	1
			1.1574	0.5508	0.04453	0.2113	[-0.407 0.422]	0.01303	11	0.04463	0.20679	[-0.357 0.454]	0.02204	2
			1.1465	0.2556	0.04537	0.213	[-0.404 0.431]	0.01313	11	0.04302	0.20451	[-0.36 0.442]	0.0218	2
			1.3209	0.2531	0.03739	0.20398	[-0.395 0.405]	0.01479	11	0.04401	0.21044	[-0.384 0.442]	0.02456	2
			1.2998	0.2591	0.03998	0.2024	[-0.399 0.305]	0.01381	11	0.04253	0.21101	[-0.38 0.447]	0.02312	1
50	90	3000	1.3313	0.2496	0.04123	0.20267	[-0.401 0.39]	0.01397	11	0.0427	0.20886	[-0.399 0.44]	0.02439	2
			1.2945	0.2596	0.04357	0.20315	[-0.403 0.394]	0.01438	10	0.04267	0.21171	[-0.381 0.449]	0.03365	1
			1.2939	0.2612	0.04525	0.20342	[-0.396 0.401]	0.01444	10	0.04782	0.21108	[-0.395 0.433]	0.02569	2

Table 13: Classification of Zoo data set based on proposed method

n	Pop.	Iter.	NSGA II		Training data set					Test data set				
			MSE	DBI	MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
			0.7917	0.3458	0.00051	0.02279	[-0.044 0.045]	0.00261	0	0.00382	0.05222	[-0.132 0.073]	0.01204	1
			0.7954	0.3307	0.00057	0.02405	[-0.047 0.047]	0.00275	1	0.0035	0.05126	[-0.126 0.075]	0.01125	1
50	40	3000	0.826	0.3252	0.00058	0.02443	[-0.048 0.049]	0.0028	1	0.00333	0.05226	[-0.113 0.092]	0.01021	1
			0.793	0.3429	0.00066	0.02602	[-0.051 0.051]	0.00298	0	0.00307	0.05212	[-0.11 0.094]	0.01182	1
			0.8924	0.3111	0.00087	0.02981	[-0.059 0.058]	0.00318	1	0.00318	0.05272	[-0.092 0.115]	0.01004	2
			0.7185	0.3212	0.00047	0.02188	[-0.042 0.044]	0.0025	0	0.00451	0.06023	[-0.104 0.132]	0.01164	1
			0.747	0.2903	0.00076	0.02783	[-0.054 0.055]	0.00319	0	0.00413	0.05987	[-0.121 0.114]	0.01097	2
70	50	3000	0.7383	0.2911	0.00082	0.02895	[-0.057 0.057]	0.00332	1	0.00304	0.05631	[-0.112 0.108]	0.01126	2
			0.7081	0.239	0.00083	0.0291	[-0.056 0.058]	0.00333	0	0.00439	0.06055	[-0.123 0.115]	0.01151	1
			0.7482	0.2816	0.00085	0.02946	[-0.058 0.058]	0.00338	1	0.0029	0.05502	[-0.108 0.107]	0.011	2
			0.6831	0.2927	0.00049	0.02233	[-0.044 0.044]	0.00256	0	0.00555	0.06215	[-0.142 0.102]	0.01613	2
			0.6834	0.2508	0.00043	0.021	[-0.041 0.041]	0.0024	0	0.00591	0.06015	[-0.135 0.101]	0.01123	2
90	60	3000	0.6969	0.1378	0.00065	0.0258	[-0.05 0.051]	0.00296	0	0.00608	0.06658	[-0.142 0.119]	0.01129	2
			0.7113	0.1317	0.00078	0.02823	[-0.055 0.056]	0.00323	1	0.00587	0.06653	[-0.146 0.115]	0.0113	2
			0.7246	0.1221	0.00028	0.01711	[-0.033 0.034]	0.00196	0	0.00261	0.0516	[-0.109 0.094]	0.01032	1

for evaluating the performance of considered mitigation method. ndnSIM simulation environment reproduces the basic structures of a NDN node (i.e., CS, PIT, FIB, strategy layer, and so on). The proposed detection method (first phase) was implemented by the MATLAB software on the Intel Pentium 2.13 GHz CPU, 4 GB RAM running Windows 7 Ultimate. This algorithm deployed to C++ project integrating as a C++ shared library using the MATLAB compiler. Then, this C++ program was integrated with ndnSIM environment to be able to adjust in the simulation environment. The proposed adaptive reaction was also implemented with C++ in ndnSIM environment. We demonstrate through simulations that our countermeasure satisfies consider-

ably applied performance metrics as compared to two recently applied DoS attack mitigation methods namely satisfaction-based pushback and satisfaction-based Interest acceptance [15]. We perform 10 times simulation runs to calculate the average performance metrics.
Our experiments are performed over two topologies shown in Figs. 9 and 10. Fig. 9 corresponds to DFN-like (Deutsche Forschungsnetz as the German Research Network) [100], and Fig. 10 corresponds to the AT&T network [101]. We use the symbols Cx, Px, Rx, and Ax to represent x-th consumer, producer, router and adversary nodes, respectively. In spite of various arguments and experiments, there is no typically and properly justification for NDN parameters and they have specified based on authors' experiences and designs [2]. The experimental setup (i.e., attack and non-attack traffics modeling) is performed over two applied topologies as follows. For attack effectiveness, we examine the performance of the network's data packet delivery and satisfied Interest rate under the different scenarios (see DoS attacks issues in section 3):

1. Interest flooding attack (dynamically-generated Interest packets) for the existent Data.
2. Interest flooding (dynamically-generated Interest packets) for the non-existent Data. It can be in the form of brute-force attack (very high distribution of Interest) or normal distribution of Interest.
3. Hijacking, in which a producer silently drops all incoming Interest traffic in downstream interfaces.
4. Content poisoning (bogus data packets), in which a producer deliberately signs data packets with a wrong key. We assume that the routers firstly check the signature filed of data packet, then cache and route the packet toward its destination if the signature is valid. Hence, the bogus data packets cannot be cached in the intermediate routers.

In our configurations, we set nodes' PIT size to 120 KB, while the Interest expiration time was set to the default timeout of 4 sec. We set the link delay and queue length parameters to fixed values for every node in the simulated topologies. In particular, we set delay and queue length to 10 ms and 400 for both considered topologies, respectively. The PIT entries replacement policy was adopted to the least-recently-used (the oldest entry with minimum number of incoming faces will be removed if PIT size reached its limit) as a widely used strategy. The nodes' cache capacity was set to 1000 contents and cache replacement policy was set to least-recently-used method. The other system settings of investigated network topologies are summarized in Table 14. As shown in this table, we ran various traffic

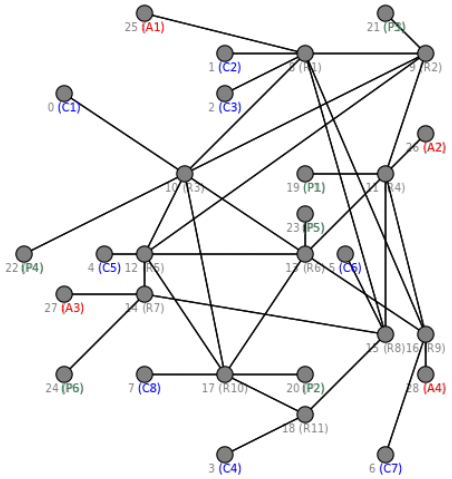


Figure 9: DFN-like topology

patterns in which each configuration changes in every 10 simulation runs in order to perform different network characteristics.
We first analyze the topologies without any adversarial traffic, then with adversarial traffic, finally consideration of the proposed mitigation method over the illegitimate traffics. Our assumption is that, the behavior of legitimate (honest) consumers is unchanged in duration of the simulation, and the adversary is not allowed to control routers. To study the performance of our proposed countermeasure algorithm under range of conditions, we varied the percentage of attackers and their run times in the considered topologies in Table 14.

11. The proposed countermeasure: proactive detection and adaptive reaction

In this section, we introduce our method, a two phases -detection and reaction- framework for mitigating DoS attacks in NDN.

11.1. Detection Phase

This step adopts our proposed intelligent classifier from section 8.1. We choose the DFN-like topology (Fig. 9) in the training phase with the recommended parameter settings in Table 14. We then apply this trained network for the detection purposes in both DFN-like and AT&T topologies.
NDN routers can easily keep track of unsatisfied (expired) Interests and use this information for DoS attack countermeasures such as, pending Interests per outgoing and incoming interfaces, and pending Interests per

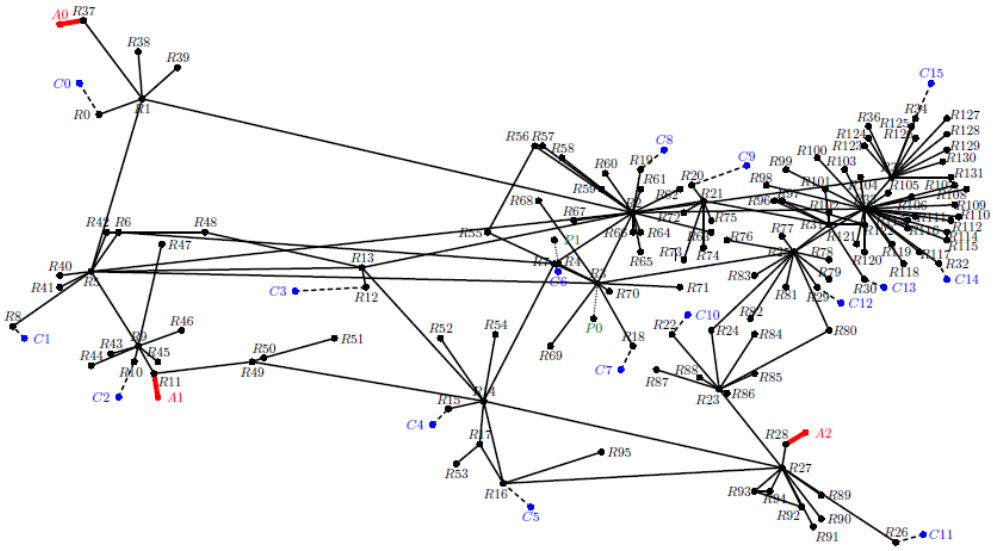


Figure 10: AT&T topology

Table 14: Network parameters considered

Node	Distribution	Pattern	Frequency	Run time (minute)	Producer	Goal
DFN-like topology (Fig. 9)						
C1	randomize	uniform	[100 500]	0-40	P1	normal
C2	randomize	exponential	[100 500]	2-40	P2	normal
C3	Zipf-Mandelbort (α = [0.5 0.9])	exponential	[100 500]	3-40	P3	normal
C4	randomize	uniform	[100 500]	4-40	P6	normal
C5	Zipf-Mandelbort (α = [0.5 0.9])	exponential	[100 500]	3-40	P2, P3	normal
C6	randomize	uniform	[100 500]	5-40	P3	normal
C7	randomize	uniform	[100 500]	7-16, 22-31	P6, P4	sign data with the wrong key
C8	randomize	exponential	[100 500]	8-18, 25-40	P1	normal
A1	randomize	uniform	[1500 3000]	7-16	P1	Interest flooding for existence data
A2	Zipf-Mandelbort (α = [0.5 0.9])	uniform	[1500 3000]	22-31	no producer	Interest flooding for non-existence data
A3	randomize	uniform	[400 800]	7-16	P5 (hijacker)	hijacking incoming Interest packets
A4	randomize	exponential	[1500 3000]	22-31	P6	Interest flooding for existence data
AT&T topology (Fig. 10)						
C0, C7	randomize	uniform	[200 600]	0-50	P0, P1	normal
C1, C8	randomize	exponential	[200 600]	2-50	P0	normal
C2, C9	randomize	exponential	[200 600]	3-50	P1	normal
C3, C10	randomize	uniform	[200 600]	4-50	P1	normal
C4, C11	Zipf-Mandelbort (α = [0.5 0.9])	exponential	[200 600]	5-50	P0, P1	normal
C5, C12, C13	randomize	uniform	[200 600]	6-50	P0, P1	normal
C6, C14, C15	randomize	uniform	[200 600]	8-50	P1	normal
A0	randomize	uniform	[1000 3000]	7-25	P1	Interest flooding for existence data
A0	randomize	exponential	[1000 3000]	30-45	P1	Interest flooding for existence data
A1	Zipf-Mandelbort (α = [0.5 0.9])	exponential	[500 1000]	7-25	P0	sign data with the wrong key
A1	Zipf-Mandelbort (α = [0.5 0.9])	uniform	[1000 3000]	30-45	no producer	Interest flooding for non-existence data
A2	randomize	exponential	[1000 3000]	7-25	no producer	Interest flooding for non-existence data
A2	randomize	uniform	[1000 3000]	30-45	P1	Interest flooding for existence data

namespace. The proper combining/choosing of statistic parameters in NDN routers for maximum effectiveness against attacks and anomalies, minimum disordering of legitimate traffics, and distinguishing between 'good' and 'bad' Interest packets are research challenges [15, 20]. Hence, we employed simple intrinsic features from the network which is shown in Table 15 (i.e., the input features in the RBF neural network). In the training process, all the features beginning with 'In' are suitable for prediction of the misbehaving consumers and the features by 'Out' are suitable for prediction of the misbehaving producers. Taking into account only a specific or a group (e.g., 'In' or 'Out') of features may cause the detection algorithm to report a wrong prediction. For example, if there are two PIT entries that share the same prefix and one Data packet arrives, there will be two entries of In/Out satisfied Interest but only one In/Out Data, since both Interests can

be satisfied with the same Data. Hence, if a number of In/Out Data be more than the In/Out satisfied Interest for a given interface or vice versa, it would not be a misbehaving. Another instance is that, Interest packets from a consumer are possible to arrive to several routers and perhaps several producers that can satisfy the Interests. Corresponding data packet will send back from producer(s). A router in the middle way, receives the first packet from any producer and will forward it to the consumer and remove the PIT entry. When the second Data object arrives to the router, it will be discarded by the routers as unsolicited. Hence, it is more likely that a rate of In/Out Data or DropData be more than In/Out Interest rate and vice versa in a corresponding interface. Obviously, it is not an attack or anomaly behavior. Also, in a given interface, the rate of the InInterest may be less that the SatisfiedInterest rate which in due to the portion of the satisfaction rate comes from the previous time interval. On the other hand, the rate of the OutData may be more than the InInterest rate, which is for routing the cached data for satisfying incoming Interest packets. To sum up, different parameters mentioned by our detection module act as weights and counterweights for misbehaving consumer and producer detection purposes.

For constructing a predictor module based on the RBF neural network, at first the centers, widths and weights are computed and adjusted using training set 75% of data set, and then the remaining part of the data set as the test set, is used to validate the trained network functionality. We trained and evaluated the network with various number of RBF units, where the three optimal results are summarized in Table 16. The optimal Pareto front solutions by NSGA II are also depicted in Fig. 11. We computed the MSE, Std., CI (95%), SEM and classification error for both training and testing parts. The histogram analysis of the classification error distribution and the regression analysis of the misclassification are shown in Figs. 12 and 13, respectively. As seen in these Figures and Table 16, third parameter settings could provide the better results as compared to the two others in terms of the applied performance metrics. Hence, these (near) optimal parameter settings are used to construct our RBF classifier (predictor).

As we expected (based on our proof in section 9), This phase constructs an optimized and more accurate RBF classifier (predictor) for our DoS attack mitigation purposes in NDN. According to the traffic flows type in the training data set (see Table 14), this predictor learned three types of traffic patterns including normal, malicious behavior from consumers and producers. This predictor module runs on routers, in order to continuously monitor per-interface required statistical information. This module is executed at fixed time intervals -typically every 0.5 sec - to provide a proactive detection behavior. Finally, based on three types of prediction (normal, misbehaving consumer and misbehaving producer), we should respond an appropriate action as detailed in the next subsection.

Table 15: Feature construction

Feature	**Description**
InInterests	a number of arrival Interest in an interface
InData	a number of arrival data in an interface
InSatisfiedInterests	a number of satisfied Interests where interface was part of the incoming set
InTimedOutInterests	a number of timed out Interests where interface was part of the incoming set
OutInterests	a number of sent Interest from an interface
OutData	a number of sent data from an interface
OutSatisfiedInterests	a number of satisfied Interests where interface was part of the outgoing set
OutTimedOutInterests	a number of timed out Interests where interface was part of the outgoing set
DropInterests	a number of dropped Interest in an interface
DropData	a number of dropped data in an interface
SatisfiedInterests	a total number of satisfied Interests
TimedOutInterests	a total number of timed out Interests

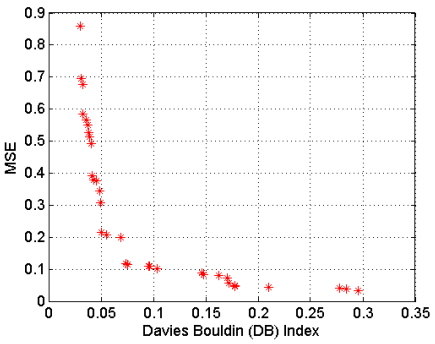
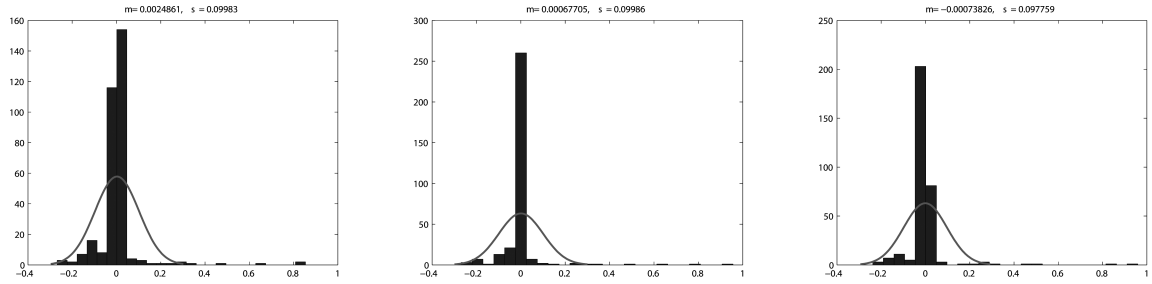


Figure 11: Optimal Pareto fronts of DFN-like training phase

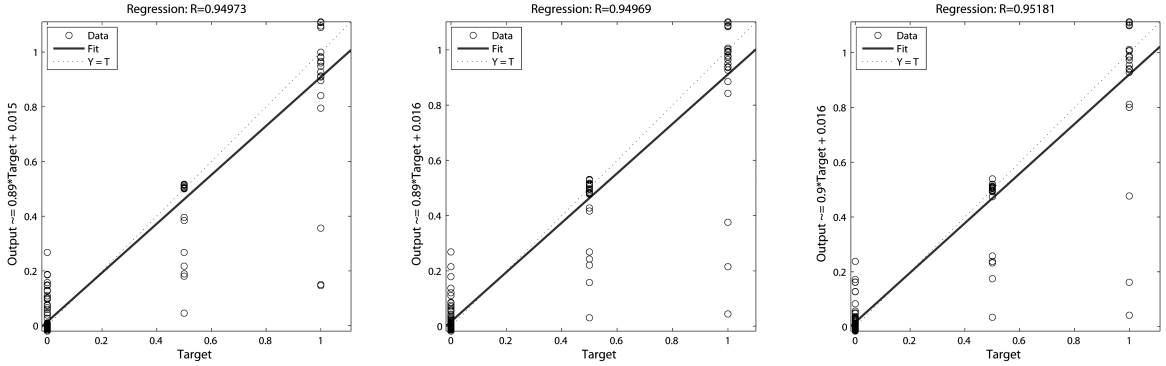
11.2. Reaction Phase

Once a DoS attack from interface j of router i has been identified with the proposed proactive detector (see section 11.1), our reaction mechanism enables and enforces explicit limitation based on the prediction type (adversary consumer or adversary producer) for each interface. The proposed intelligent proactive detector reports misbehaving in the early stages of beginning DoS attacks. Our adaptive reaction criterion for misbehaving consumer directly depends on the local interface's Interest unsatisfied ratio and for misbehaving producer directly depends on the forwarding strategy. The original settings and Interest rate are restored once the de-



(a) 1st histogram (b) 2nd histogram (c) 3rd histogram

Figure 12: The histogram analysis of the classification error distribution in DFN-like topology



(a) 1st Regression (94.97%) (b) 2nd Regression (94.96%) (c) 3rd Regression (95.18%)

Figure 13: Regression of the classification error between target and predicted output in DFN-like topology

tector module reports the normal traffic in the next time interval.

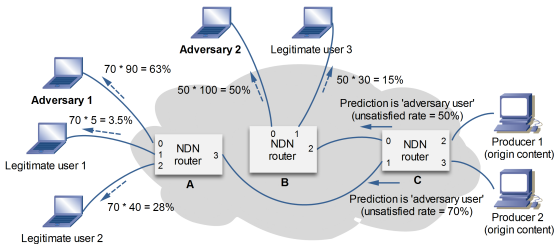


Figure 14: unsatisfied-based pushback example

11.2.1. reaction regarding to misbehaving consumer

When the proposed intelligent detector module in router detects adversarial traffics from a set of interfaces, it sends an alert message on each of them. An alert message is an unsolicited content packet which belongs to a reserved namespace (”`/pushbackmessage/alert/`”) in our implementation. There are two reasons for using content packet rather than Interest packet for carrying pushback message [11]:

1. during an attack, the PIT of next hop connected to the offending interface may be full, and therefore the alert message may be discarded, and
2. content packets are signed, while Interests are not. This allows routers to receive the content packets as a legitimate packet for processing.

The payload of an alert message contains the timestamp corresponding to the generation time of the alert message, the new reduced (unsatisfied) rate and the wait time of reduction period. The formal definition of our unsatisfied-based pushback mechanism presents in Fig. 14. Assuming in a time interval in router `C` the predictor reports a misbehaving traffic from a consumer (neigh-

Table 16: Classification of NDN data set based on proposed method

n	Pop.	Iter.	NSGA II		Training data set					Test data set				
			MSE	DBI	MSE	Std.	CI (95%)	SEM	Cls. err.	MSE	Std.	CI (95%)	SEM	Cls. err.
			0.0314	0.0979	0.00998	0.09983	[-0.192 0.197]	0.00555	2	0.02354	0.15414	[-0.301 0.304]	0.01476	3
	80	40	2500	0.0643	0.0979	0.00994	0.09986	[-0.186 0.206]	0.00555	2	0.02486	0.15841	[-0.324 0.297]	0.01517	3
				0.0315	0.0914	0.00952	0.09505	[-0.187 0.186]	0.00543	1	0.02311	0.15271	[-0.299 0.3]	0.01421	1

bor node). Also, an unsatisfied rate is 50% for `eth0` and 70% for `eth1`. Our proposed reaction mechanism is as follows:

1. Router `C` will send a pushback alert message to the neighbors from `eth0` and `eth1`.
2. Routers `A` and `B`, after receiving alert message from `C` will readjust their local interfaces limit to `'announced reduced rate'` × `'local unsatisfied rate'` in each local interface. If the new limit in the corresponding interface exceeds the predefined threshold ϕ, the corresponding interface gets new reduction of Interest rate in downstream. For instance, we assume ϕ = 5% so that router `B` decreases the Interest rate of `eth0` to 50% and `eth1` to 15%. Router `A` decreases the Interest rate in its three interfaces to 63%, 0 (the new limit rate (=3.5%) is under predefined threshold (=5%) and will not be changed) and 28% in `eth0`, `eth1` and `eth2`, respectively. This threshold allows bandwidth usage be consumed for legitimate traffics in the nearest next time interval and intensifies Interest rate reduction for adversaries in each next time intervals.
3. Our wait time strategy for the reduction period in neighbor nodes is an ascending penalty. If in a time interval *t* in interface *j* the misbehaving traffic be reported, a counter sets to 1 sec. If in the next time interval *t+1* the misbehaving again be reported, a counter sets to 2 sec. Our ascending penalty method is in $2^{counter}$. Initially, $counter = 0$ and increase linearly in each time interval. The counter is set to the initial value when there is no misbehaving prediction in the next time interval. This ascending penalty intensifies the penalty for adversaries and alleviates the bandwidth usage for legitimate (honest) users.
4. Any neighbor node may obey (ignore) the announced limit rate and send Interest packets without any restriction from the upstream interface. Our algorithm after twice refusing the alert message will band the incoming Interest packets from the corresponding interface for a long time period.

At the next iteration of the unsatisfied-based pushback mechanism, legitimate user(s) will be able to gradually improve their satisfaction rate and sending Interest packets on both router `A` and `B`. After applying the alert message in router `A`, the Interest rate of the adversary will be decreased to around 63% in the next iteration. It allows bandwidth usage be consumed for 2nd legitimate user, that it will considerably led to the increasing of legitimate Interests rate. If the adversary continues its misbehaving in the next times, the ascending wait time strategy will increase the penalty rate of the illegal Interest packets. Hence, `eth1` and `eth2` interfaces in router `A` will get through and return Data, eventually resulting in a full allowance in the link between the routers `A` and `C`.

The Pseudo-code of the unsatisfied-based pushback

mechanism is shown in Algorithm 1. In this algorithm, the `Decrease` function decreases the Interest rate from corresponding interface with announced parameters. After normal traffic prediction, the `Increase` function sets the default Interest rate on the corresponding interface in the next time interval. The `IsFresh` function checks the freshness of the alert message when there is no previously alert message.

Input: `AlertMsg`, `timestamp` of alert generation, `reduced rate` and `wait time` from interface j in router i (r_i^j)
Result: (1) adaptive pushback reaction and (2) pushback alert message generation

$counter_j = 0$ // initial counter for generating wait time in interface j
$\phi = 5\%$ // reduction threshold of Interest rate
/* section: adaptive pushback reaction */
if *AlertMsg is Pushback alert message* **then**
    if `Verify(`*AlertMsg.signature*`)` **and** `IsFresh(`*AlertMsg.timestamp*`)` **then**
        /* Pushback reaction */
        foreach *local interface j* **do**
            new rate = unsatisfied rate of j × announced reduced rate ;
            if $\phi <$ *new rate* **then**
                /* intensify the penalty */
                Decrease(*interface j, new rate, announced wait time*);
            else
                /* reset to original setting */
                Increase(*interface j, original rate*) ;
else
    Drop(*AlertMsg*) ;
    return ;
/* section: Pushback alert message generation */
if *(predictor module reports the adversary consumer (neighbor) in local interface j)* **then**
    if *(time from last sent AlertMsg to local interface* $r_i^j <$ *current local time)* **then**
        /* Pushback alert message generation */
        new time interval = $2^{counter_j}$;
        `AlertMsg` = (current `timestamp` of alert generation, current `unsatisfied rate` in local interface j, new time interval);
        `Send(`*AlertMsg to* r_i^j`)`;
        $counter_j = counter_j + 1$;
else
    /* reset to original setting */
    $counter_j = 0$;
    Increase(*interface j, original rate*) ;

Algorithm 1: Unsatisfied-based pushback algorithm

11.2.2. reaction regarding to misbehaving producer

If the predictor module predicts a misbehaving producer from an interface j, we build an adaptive and simple forwarding strategy. The main goal is to retrieve data via the best performance path(s), and to quickly recover packet delivery problem by the other (possible) legitimate producers. When a predictor module in a router i reports a misbehaving producer in an interface j, the interface status changes to the *unavailable* (can not bring data back) and will be deactivated for a predefined time interval. This type of forwarding strategy can increase the data retrieving chance for awaiting Interest packets in the PIT table by changing the forwarding path. We apply the wait time strategy from the misbehaving consumer section (see section 11.2.1). After normal prediction in the next time intervals, the interface status changes to *available* (can bring data back). It means, it is ready for forwarding Interest packets via this interface. It is expected that in the next time intervals, when there is no any legitimate producer to satisfy the corresponding Interest packets in an interface j, the predictor module reports misbehaving consumer (neighbor) from upstream interface j, where Interest packets are susceptible to be illegal traffics. Then, the rate of incoming Interest packets should gradually decrease in upstream interface j based on our ascending penalty mechanism in previous subsection.

12. Experimental results and evaluation

In this section we report the experimental evaluation of countermeasures presented in Section 11. Our countermeasures are tested over two considered topologies in Figs. 9 and 10. Each router implements the proposed detection technique discussed in Section 11.1 and adaptive reaction technique discussed in 11.2.
We report the results based on the five conditions: baseline, attack (no countermeasure), our proposed method, and two DoS mitigation methods applied in this work including satisfaction-based pushback and satisfaction-based Interest acceptance from [15]. Figs. 15 and 18 show the average Interest satisfaction ratio for legitimate users within 10 runs in DFN-like and AT&T topologies, respectively. Our results show that the proposed intelligent hybrid algorithm (proactive detection and adaptive reaction) is very effective for shutting down the adversary traffics and preventing legitimate users from service degradation by the accuracy more than 90% during the attack.
Figs. 16 and 19 demonstrate the average PIT usage within 10 runs in five considered conditions in DFN-like

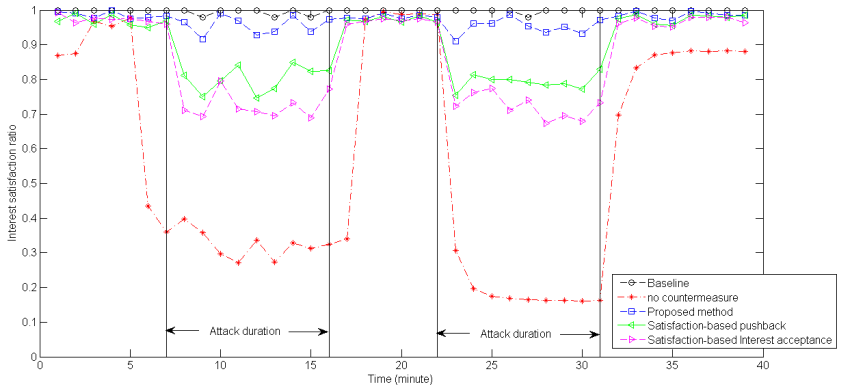


Figure 15: Interest satisfaction ratio for legitimate users in DFN

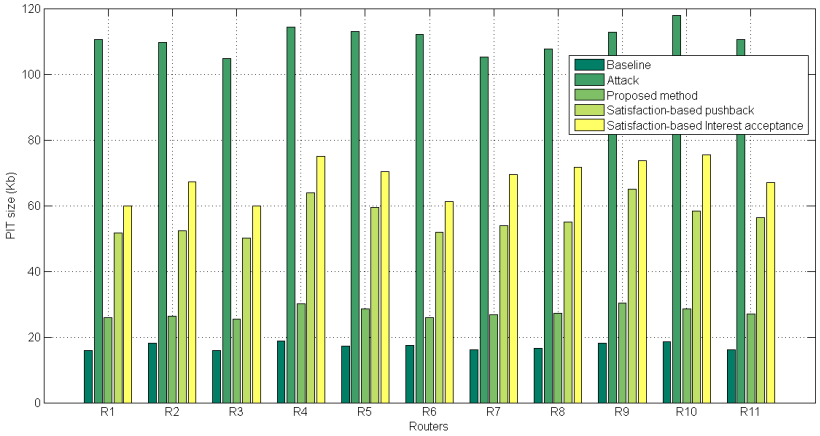


Figure 16: PIT usage with countermeasures in DFN

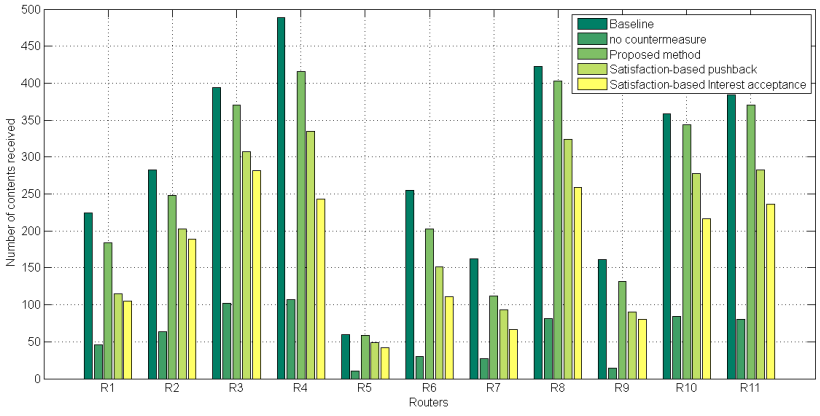


Figure 17: Effects of countermeasures in DFN (Throughput)

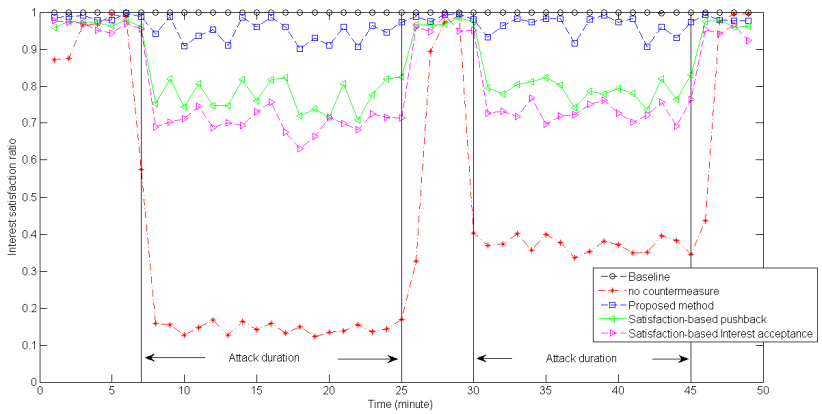


Figure 18: Interest satisfaction ratio for legitimate users in AT&T

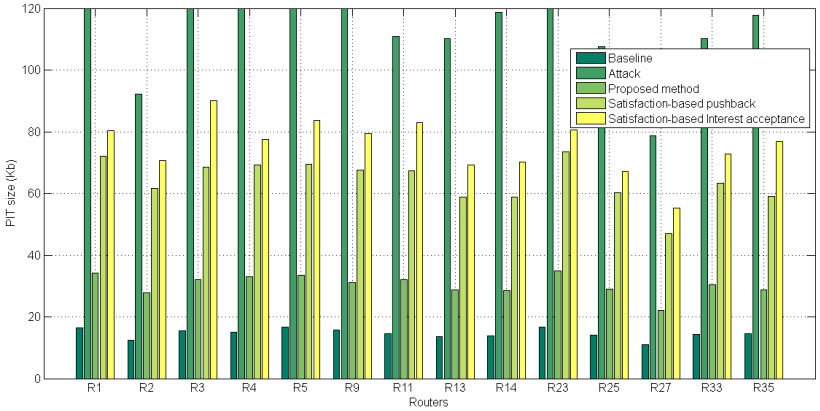


Figure 19: PIT usage with countermeasures in AT&T

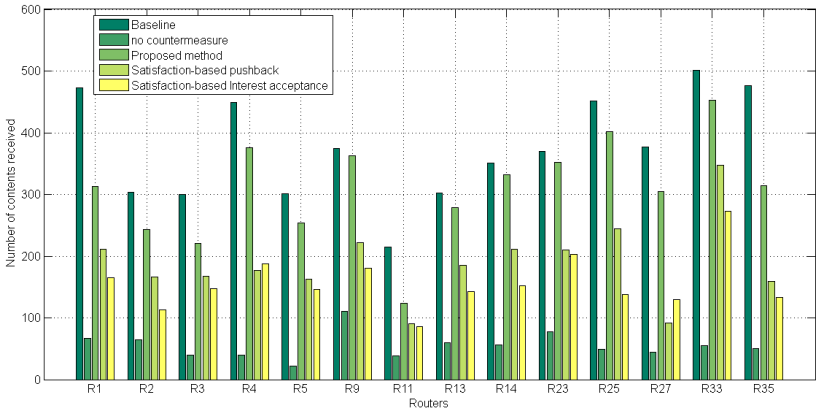


Figure 20: Effects of countermeasures in AT&T (Throughput)

and AT&T topologies, respectively. Our results show that there is a significant benefit of the proposed countermeasure in reduction of PIT usage in presence of an adversary. In Figs. 17 and 20 we show the average number of content received (throughput) in DFN-like and AT&T topologies, respectively. The results show that the proposed countermeasure is effective and efficient in presence of adversary. For clarity, we just report measurements for those routers that are affected by the attacks for AT&T topology. The most routers in both considered topologies exhibit an interesting behavior. The proposed mechanism in both steps (detection by an intelligent hybrid method and reaction by enforcing explicit limitations against adversaries) offers visibly promising performance in presence of adversary.

12.1. Two facts of DoS/DDoS mitigation in NDN

Experimental results and analysis show that the two conditions can cause a DoS mitigation method degrades service to legitimate consumers:

1. Producers can misbehave by dropping incoming Interest packets or signing data packets with the wrong keys as they are unwilling to forward data packets to legitimate consumers. We conducted this experiment in AT&T topology by producer P0 between 7-25 seconds of simulation run (see Table 14), in which consumers C0, C1, C4, C5, C7, C8, C11, C12, and C13 request their desirable data packets from that producer. When our proposed predictor in an interface reports a misbehaving producer, the corresponding interface status changes to unavailable and will be deactivated (see section 11.2.2). Consequently, the data retrieving chance for awaiting Interest packets increases by changing the forwarding path towards the producer P1 except C1 and C8, because other consumers can be satisfied with more than one producer. It is an expected behavior from a predictor until there is either no misbehaved producer or an extra well-behaved producer. As a result of this condition, it seems reasonable to decrease the rate of Interest packets for legitimate consumers.
2. A DoS mitigation technique should be able to detect malevolent behaviors and any deviation ideally long before the destructive traffics build-up and block the network traffics belonging to the attackers without denying services to legitimate consumers in a timely manner. If a mitigation technique cannot detect DoS attacks in a timely manner, the generated overload by DoS attacks prevents the resource from responding to legitimate traffic, or slows its response so significantly a (high) percentage of the legitimate Interest packets are completely disrupted. In this way, DoS mitigation techniques often create false positives (false alarms) by dropping legitimate Interest packets or enforcing limitations incorrectly against legitimate consumers. False positive refers to normal traffics when are incorrectly decreased by enforcing explicit limitations from our proposed unsatisfied-based pushback mechanism and other considered countermeasures during DoS/DDoS attacks (see section 11.2). Table 17 demonstrates the average rate of false positives obtained by our method and other applied countermeasures within 10 runs. This table shows that the proposed mitigation method is characterized by an extremely low false positive rate as compared to other countermeasures which is important when dealing with DoS/DDoS attacks. It can be concluded that the proposed intelligent hybrid predictor is able to detect DoS/DDoS attacks in a timely manner to prevent service degradation for legitimate users.
   A future work is needed in the classification of legitimate users' traffics as either good (non-malicious), bad (malicious) or low and high prone to attack traffics (non-malicious, but with the same properties as malicious traffics).

12.2. Discussion

A new intelligent hybrid algorithm (**proactive detection and adaptive reaction**) for mitigating DoS attacks in Named Data Networking has been proposed. The first part (**detection**) of this new algorithm (Fig. 4) consists of two phases: training/optimization and prediction/classification. In the training phase, an hybrid optimization algorithm has been developed to resolve the hybrid learning problem of RBF neural networks using multiobjective evolutionary algorithm and PSO. The first step of this phase adjusts RBF units' centers based on NSGA II through two conflicting objectives: well-separated centers (by Davies Boulding Index (DBI)) and local optimization of centers (by Mean-Squared Error (MSE)). Second step of this phase trains units' widths and output weights using PSO. This step tunes and adjusts widths and weights simultaneously by the well-separated centers from the previous step. In the prediction phase, a simple and an effective prediction algorithm has been designed to classify the new input patterns in their actual classes. This part of our hybrid algorithm has been successfully applied to define a more accurate RBF classifier over the NDN traffic flows as

Table 17: Comparison of false positive rate (mean of 10 runs)

Topology	No countermeasure	Satisfaction-based Interest acceptance	Satisfaction-based pushback	Our method
DFN	59.78%	21.05%	14.47%	6.44%
AT&T	66.43%	24.29%	19.13%	9.26%

well as distinguish intelligently DoS attack traffics.
Convergence of the proposed RBF classifier (predictor) is studied for finding global and optimal classification of different benchmarking data sets as Wine, Iris, Ionosphere and Zoo. We applied the single-objective approach in Tables 2-5 (training units' centers) and Tables 6-9 (training units' widths, output weights and calculating the misclassification error), and our conflicting two-objective approach in Figs. 5-8 (Pareto front of the units' centers solutions) and Tables 10-13 (training units' widths, output weights and calculating the misclassification error). Experimental results confirm the accuracy and the robustness of the proposed approach based on the several performance metrics: MSE, Standard Deviation (Std.), Standard Error of Mean (SEM), Confidence Interval (CI 95%) and the number of incorrect classification.
The feasibility and efficiency of the proposed RBF classifier (predictor) method was compared to four well-known and frequently used optimization algorithms. Tables 6-9 demonstrate the final results, using PSO, Genetic Algorithm (GA), Imperialist Competitive Algorithm (ICA) and Differential Evolution (DE). The proposed algorithm in this paper outperforms all applied methods based on the (near) optimal results in the number of correct classification, MSE and Std. criteria. It can be concluded that the proposed intelligent hybrid algorithm is able to construct more accurate and well-tuned RBF classifier for (near) optimal classification of input patterns.
Although, the proposed method and other methods use different parameter settings. Our method was repeated 5 times and others were repeated 20 times independently to find the global results in the training/optimization phase; therefore, the effect of tuning parameters on performance of the methods are studied. We repeated the proposed training phase less than other methods to show that our two-objective approach is able to tune and adjust RBF parameters faster and more accurate than other methods.
The proposed intelligent classifier was successfully adopted in the detection phase of our countermeasure (see section 11.1). After constructing the intelligent hybrid classifier (predictor) module, an **adaptive reaction** mechanism by enforcing explicit limitations against adversaries was proposed to mitigate potential DoS/DDoS attacks in NDN (see section 11.2). Finally, convergence, feasibility and efficiency of the proposed algorithm (proactive detection and adaptive reaction) is studied for finding the optimal placement of RBF units' centers, units' widths and output weights and measuring the suitable performance over two network topologies including DFN-like (Fig. 9) and AT&T (Fig. 10). The results were promising as compared to two recently proposed DoS mitigation methods from [15] based on the average of Interest satisfaction ratio for legitimate users, the PIT usage, the number of received contents (throughput), and a very low false positive rate over 10 simulation runs.

13. Conclusion

NDN is a newly proposed future Internet architecture which it is important to address its resilience in face of DoS/DDoS attacks. We examined the most current instances of DoS/DDoS attacks to show that an adversary with limited resources can serve service degradation for legitimate users. We then introduced our intelligent hybrid algorithm for proactive detection of DoS attacks and adaptive reaction for mitigating. In the detection phase, a combination of multiobjective evolutionary optimization and RBF neural network has been applied. This approach consists of two phases: training/optimization and prediction/classification. In the training phase, we investigate the implementation of a multiobjective approach and PSO in the design of RBF neural network in order to improve the accuracy of classification problems. We apply NSGA II to determine the Pareto solutions of RBF units' centers in terms of the well-separated centers through DBI and their local optimization through MSE. Then, the optimization and tuning of the units' widths and output weights are accomplished by using the PSO, where the each particle encodes a set of widths and weights. Moreover, the structure of this step is simple and easy to implement, yet very effective in terms of several performance criteria. In the prediction phase, we employ a simple algorithm to classify efficiency the new input patterns with the minimum misclassification error. This hybrid algorithm was applied on four benchmarking data sets to

verify the algorithm accuracy and robustness in classification problems.
Subsequently, after constructing a more accurate classifier (detector), we performed a simple adaptive reaction algorithm by enforcing explicit limitations against adversaries which was very effective and efficient for shutting down the attackers with the robust recovery from network failures and accuracy more than 90% in terms of the average of Interest satisfaction ratio for legitimate users, the PIT usage, the number of received contents (throughput), and a very low false positive rate over 10 simulation runs.
We are currently investigating inter-domain DoS attacks. We leave further investigations to future work.

14. Acknowledgment

This work was partially supported by projects TIN2013-47272-C2-2 and SGR-2014-881.

References

[1] Conti, M., Gasti, P., Teoli, M.. A lightweight mechanism for detection of cache pollution attacks in named data networking. Computer Networks 2013;57(16):3178–3191.
[2] Rossini, G., Rossi, D.. Evaluating ccn multi-path interest forwarding strategies. Computer Communications 2013;36(7):771–778.
[3] Li, C., Liu, W., Okamura, K.. A greedy ant colony forwarding algorithm for named data networking. In: Proceedings of the Asia-Pacific Advanced Network; vol. 34. 2012, p. 17–26.
[4] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., Ohlman, B.. A survey of information-centric networking (draft). In: Information-Centric Networking. Dagstuhl Seminar Proceedings; Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany; 2011,.
[5] Lee, H., Nakao, A.. User-assisted in-network caching in information-centric networking. Computer Networks 2013;57(16):3142–3153.
[6] Smetters, D.K., Jacobson, V.. Securing network content. parc tr-2009-1. Tech. Rep.; October, 2009.
[7] Ran, J., Lv, N., Zhang, D., Ma, Y., Xie, Z.. On performance of cache policies in named data networking. In: International Conference on Advanced Science and Electronics Information (ICACSEI). Atlantis Press; 2013, p. 668–671.
[8] Amadeo, M., Campolo, C., Molinaro, A., Ruggeri, G.. Content-centric wireless networking: A survey. Computer Networks 2014;72:1–13.
[9] Dannewitz, C., Kutscher, D., Ohlman, B., Farrell, S., Ahlgren, B., Karl, H.. Network of information (netinf) - an information-centric networking. Computer Communications 2013;36(7):721–735.
[10] Jiang, X., Bi, J.. Technical report: Named content delivery network. Tech. Rep.; 2013.
[11] Carofiglio, G., Gallo, M., Muscariello, L.. On the performance of bandwidth and storage sharing in information-centric networks. Computer Networks 2013;57(17):3743–3758.
[12] Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.. Networking named content. In: Proceedings of the 5th international conference on Emerging networking experiments and technologies. CoNEXT '09; New York, NY, USA: ACM; 2009,.
[13] M. Xie, I.W., Wang, H.. Enhancing cache robustness for content-centric networking. In: INFOCOM12. 2012, p. 2426–2434.
[14] Bari, M., Chowdhury, S., Ahmed, R., Boutaba, R., Mathieu, B.. A survey of naming and routing in information-centric networks. In: Communications Magazine, IEEE; vol. 50. 2012, p. 44–53.
[15] Afanasyev, A., Mahadevan, P., Moiseenko, I., Uzun, E., Zhang, L.. Interest flooding attack and countermeasures in named data networking. In: IFIP Networking Conference, 2013. 2013, p. 1–9.
[16] Karami, A.. Data clustering for anomaly detection in content-centric networks. International Journal of Computer Applications 2013;81(7):1–8. Published by Foundation of Computer Science, New York, USA.
[17] Wählisch, M., Schmidt, T.C., Vahlenkamp, M.. Backscatter from the data plane — threats to stability and security in information-centric networking. CoRR 2012;abs/1205.4778.
[18] Choi, S., Kim, K., Kim, S., hee Roh, B.. Threat of dos by interest flooding attack in content-centric networking. In: International Conference on Information Networking (ICOIN). 2013, p. 315–319.
[19] Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J.D., Smetters, D.K., et al. Named data networking (ndn) project. In: In Proceedings of the ACM SIGCOMM workshop on Information-centric networking. PARC TR-2010-3; 2010, p. 68–73.
[20] Gasti, P., Tsudik, G., Uzun, E., Zhang, L.. Dos and ddos in named-data networking. In: 22nd International Conference on Computer Communications and Networks (ICCCN 2013). 2013,.
[21] Compagno, A., Conti, M., Gasti, P., Tsudik, G.. Poseidon: Mitigating interest flooding ddos attacks in named data networking. CoRR 2013;abs/1303.4823.
[22] Oke, G., Loukas, G., Gelenbe, E.. Detecting denial of service attacks with bayesian classifiers and the random neural network. In: International Fuzzy Systems Conference, FUZZ-IEEE 2007. 2007, p. 1–6.
[23] Fiore, U., Palmieri, F., Castiglione, A., Santis, A.D.. Network anomaly detection with the restricted boltzmann machine. Neurocomputing 2013;122:13–23. Advances in cognitive and ubiquitous computing Selected papers from the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012).
[24] Liao, H.J., Lin, C.H.R., Lin, Y.C., Tung, K.Y.. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications 2013;36(1):16–24.
[25] Kolias, C., Kambourakis, G., Maragoudakis, M.. Swarm intelligence in intrusion detection: A survey. Computers and Security 2011;30(8):625–642.
[26] Han, H.G., li Chen, Q., Qiao, J.F.. An efficient self-organizing rbf neural network for water quality prediction. Neural Networks 2011;24:717–725.
[27] Shahreza, M.L., Moazzami, D., Moshiri, B., Delavar, M.. Anomaly detection using a self-organizing map and particle swarm optimization. Scientia Iranica 2011;18(6):1460–1468.
[28] Li, X.L., Jia, C., Liu, D.X., Ding, D.W.. Nonlinear adaptive control using multiple models and dynamic neural networks. Neurocomputing 2014;136:190–200.
[29] Michailidis, E., Katsikas, S., Georgopoulos, E.. Intrusion

detection using evolutionary neural networks. In: Informatics, 2008. PCI '08. Panhellenic Conference on. 2008, p. 8–12.

[30] Jiang, X., Liu, K., Yan, J., Chen, W.. Application of improved {SOM} neural network in anomaly detection. Physics Procedia 2012;33:1093–1099.

[31] Wang, G., Hao, J., Ma, J., Huang, L.. A new approach to intrusion detection using artificial neural networks and fuzzy clustering. Expert Systems with Applications 2010;37(9):6225–6232.

[32] Gan, M., Peng, H., ping Dong, X.. A hybrid algorithm to optimize rbf network architecture and parameters for nonlinear time series prediction. Applied Mathematical Modelling 2012;36:2911–2919.

[33] Zhang, Z., Wang, T., Liu, X.. Melt index prediction by aggregated {RBF} neural networks trained with chaotic theory. Neurocomputing 2014;131:368–376.

[34] Gan, M., Peng, H.. Stability analysis of rbf network-based state-dependent autoregressive model for nonlinear time series. Applied Soft Computing 2012;12(1):174–181.

[35] Lee, C.M., Ko, C.N.. Time series prediction using rbf neural networks with a nonlinear time-varying evolution pso algorithm. Neurocomputing 2009;73(1-3):449–460.

[36] Tsekouras, G.E., Tsimikas, J.. On training rbf neural networks using input-output fuzzy clustering and particle swarm optimization. Fuzzy Sets Systems 2013;221:65–89.

[37] Qasem, S.N., Shamsuddin, S.M.. Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis. Applied Soft Computing 2011;11(1):1427–1438.

[38] Sieniutycz, S., JeÅowski, J.. 1 - brief review of static optimization methods. In: Energy Optimization in Process Systems and Fuel Cells (Second Edition). Amsterdam: Elsevier; second edition ed.; 2013, p. 1–43.

[39] Sun, Y., Zhang, L., Gu, X.. A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. Neurocomputing 2012;98(3):76–89.

[40] Karami, A., Johansson, R.. Choosing dbscan parameters automatically using differential evolution. International Journal of Computer Applications 2014;91(7):1–11. Published by Foundation of Computer Science, New York, USA.

[41] Du, H., Zhang, N.. Time series prediction using evolving radial basis function networks with new encoding scheme. Neurocomputing 2008;71(7-9):1388–1400.

[42] Fathi, V., Montazer, G.A.. An improvement in {RBF} learning algorithm based on {PSO} for real time applications. Neurocomputing 2013;111:169–176.

[43] Montazer, G.A., Khoshniat, H., Fathi, V.. Improvement of {RBF} neural networks using fuzzy-osd algorithm in an online radar pulse classification system. Applied Soft Computing 2013;13(9):3831–3838.

[44] Tsekouras, G.E.. A simple and effective algorithm for implementing particle swarm optimization in rbf network's design using input-output fuzzy clustering. Neurocomputing 2013;108:36–44.

[45] Kokshenev, I., Braga, A.P.. A multi-objecitve approach to rbf network learning. Neurocomputing 2008;71:1203–1209.

[46] Karami, A., Guerrero-Zapata, M.. A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks. Neurocomputing 2014;doi: http://dx.doi.org/10.1016/j.neucom.2014.08.070.

[47] Kärkkäinen, I., Fränti, P.. Minimization of the value of davies-bouldin index. In: Proceedings of the LASTED International Conference signal proccessing and communications. Marbella, Spain; 2000, p. 426–432.

[48] Kuo, R.J., Syu, Y.J., Chen, Z.Y., Tien, F.C.. Integration of particle swarm optimization and genetic algorithm for dynamic clustering. Informaton Sciences 2012;195:124–140.

[49] Jin, Y.. Multi-Objective Machine Learning. Studies in Computational Intelligence; Springer; 2006. ISBN 9783540306764.

[50] Etghani, M.M., Shojaeefard, M.H., Khalkhali, A., Akbari, M.. A hybrid method of modified nsga-ii and topsis to optimize performance and emissions of a diesel engine using biodiesel. Applied Thermal Engineering 2013;59:309–315.

[51] Chang, P.C., Chen, S.H.. The development of a sub-population genetic algorithm ii (spga ii) for multi-objective combinatorial problems. Applied Soft Computing 2009;9(1):173–181.

[52] Mert, S.O., Özçelik, Z., Özçelik, Y., Dinçer, I.. Multi-objective optimization of a vehicular {PEM} fuel cell system. Applied Thermal Engineering 2011;31(13):2171–2176.

[53] Zangooei, M.H., Habibi, J., Alizadehsani, R.. Disease diagnosis with a hybrid method {SVR} using nsga-ii. Neurocomputing 2014;136:14–29.

[54] Lefort, V., Knibbe, C., Beslon, G., Favrel, J.. Simultaneous optimization of weights and structure of an rbf neural network. In: Proceedings of the 7th international conference on Artificial Evolution. EA'05; Berlin, Heidelberg: Springer-Verlag; 2006, p. 49–60.

[55] Kokshenev, I., Braga, A.P.. An efficient multi-objective learning algorithm for rbf neural network. Neurocomputing 2010;73:2799–2808.

[56] Qasem, S.N., Shamsuddin, S.M., Hashim, S.Z.M., Darus, M., Al-Shammari, E.. Memetic multiobjective particle swarm optimization-based radial basis function network for classification problems. Information Sciences 2013;239:165–190.

[57] Carlisle, A., Dozier, G.. An off-the-shelf pso. In: Proceedings of the Particle Swarm Optimization Workshop. 2001, p. 16.

[58] Kennedy, J., Eberhart, R.C.. Swarm Intelligence. Morgan Kaufmann, San Francisco, CA; 2001.

[59] Urade, H.S., Patel, R.. Dynamic particle swarm optimization to solve multi-objective optimization problem. Procedia Technology 2012;6:283–290.

[60] Yi, C., Afanasyev, A., Wang, L., Zhang, B., Zhang, L.. Adaptive forwarding in named data networking. SIGCOMM Computer Communications Rev 2012;42(3):62–67.

[61] Yi, C., Afanasyev, A., Moiseenko, I., Wang, L., Zhang, B., Zhang, L.. A case for stateful forwarding plane. Computer Communications 2013;36(7):779–791.

[62] Lauinger, T.. Security & scalability of content-centric networking. 2010.

[63] Dai, H., Wang, Y., Fan, J., Liu, B.. Mitigate ddos attacks in ndn by interest traceback. In: 2nd IEEE International Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN 2013). Turin, Italy; 2013,.

[64] Broomhead, D.S., Lowe, D.. Multivariable functional interpolation and adaptive networks. Complex Systems 1998;2:321–355.

[65] Hamad, A., Yu, D., Gomm, J.B., Sangha, M.S.. Radial basis function neural network in fault detection of automotive engines. In: International Journal of Engineering, Science and Technology; vol. 2. 2010, p. 1–8.

[66] Neruda, R., Kudová, P.. Learning methods for radial basis function networks. Future Generation Computer Systems 2005;21(7):1131–1142.

[67] Tan, K.K., Tang, K.Z.. Adaptive online correction and interpolation of quadrature encoder signals using radial basis functions. IEEE Transactions on Control Systems Technology 2005;13(3):370–377.

[68] Bai, Y., Zhang, L.. Genetic algorithm based self-growing

training for rbf neural networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN '02); vol. 1. 2002, p. 840–845.

[69] Kennedy, J., Eberhart, R.. Particle swarm optimization. In: Proceedings in IEEE International Conference Neural Networks; vol. 4. 1995, p. 1942–1948.

[70] Hong, X., Gao, J., Chen, S., Harris, C.J.. Particle swarm optimisation assisted classification using elastic net prefiltering. Neurocomputing 2013;122:210–220.

[71] Eberhart, R.C., Shi, Y.. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the Evolutionary Computation; vol. 1. 2000, p. 84–88.

[72] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation 2002;6(2):182–197.

[73] Yusoff, Y., Ngadiman, M.S., Zain, A.M.. Overview of nsga-ii for optimizing machining process parameters. Procedia Engineering 2011;15:3978–3983.

[74] Fallah-Mehdipour, E., Haddad, O.B., Tabari, M.M.R., Mario, M.A.. Extraction of decision alternatives in construction management projects: Application and adaptation of nsga-ii and {MOPSO}. Expert Systems with Applications 2012;39(3):2794–2803.

[75] Chen, J., Ren, Z., Fan, X.. Particle swarm optimization with adaptive mutation and its application research in tuning of pid parameters. In: Proc. 1st International Symposium on Systems and Control in Aerospace and Astronautics. 2006, p. 990–994.

[76] Azad, S.K., Azad, S.K., Kulkarni, A.J.. Structural optimization using a mutation-based genetic algorithm. International journal of optimization in civil engineering 2012;2(1):80–100.

[77] Karami, A., Johansson, R.. Utilization of multi attribute decision making techniques to integrate automatic and manual ranking of options. Journal of Information Science and Engineering 2014;30(2):519–534.

[78] Davies, D.L., Bouldin, D.W.. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1979;PAMI-1(2):224–227.

[79] Asuncion, A., Newman, D.. UCI machine learning repository. 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[80] Shokrian, M., High, K.A.. Application of a multi objective multi-leader particle swarm optimization algorithm on {NLP} and {MINLP} problems. Computers & Chemical Engineering 2014;60:57–75.

[81] Li, N.J., Wang, W.J., Hsu, C.C.J., Chang, W., Chou, H.G., Chang, J.W.. Enhanced particle swarm optimizer incorporating a weighted particle. Neurocomputing 2014;124:218–227.

[82] Khare, A., Rangnekar, S.. A review of particle swarm optimization and its applications in solar photovoltaic system. Applied Soft Computing 2013;13(5):2997–3006.

[83] Yazdani, D., Nasiri, B., Sepas-Moghaddam, A., Meybodi, M.R.. A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. Applied Soft Computing 2013;13(4):2144–2158.

[84] Thida, M., Eng, H.L., Monekosso, D.N., Remagnino, P.. A particle swarm optimisation algorithm with interactive swarms for tracking multiple targets. Applied Soft Computing 2013;13(6):3106–3117.

[85] Marinakis, Y., Iordanidou, G.R., Marinaki, M.. Particle swarm optimization for the vehicle routing problem with stochastic demands. Applied Soft Computing 2013;13(4):1693–1704.

[86] Espezua, S., Villanueva, E., Maciel, C.D.. Towards an efficient genetic algorithm optimizer for sequential projection pursuit. Neurocomputing 2014;123:40–48.

[87] Tsai, C.W., Shih, C., Wang, J.R.. Using genetic algorithms to calibrate the user-defined parameters of {IIST} model for {SBLOCA} analysis. Annals of Nuclear Energy 2014;63:499–505.

[88] Faraji, R., Naji, H.R.. An efficient crossover architecture for hardware parallel implementation of genetic algorithm. Neurocomputing 2013;128:316–327.

[89] Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.. A novel approach for traffic accidents sanitary resource allocation based on multi-objective genetic algorithms. Expert Systems with Applications 2013;40(1):323–336.

[90] Thakur, M.. A new genetic algorithm for global optimization of multimodal continuous functions. Journal of Computational Science 2014;5(2):298–311.

[91] Köker, R.. A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. Information Sciences 2013;222:528–543.

[92] Atashpaz-Gargari, E., Lucas, C.. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE; 2007, p. 4661–4667.

[93] Goldansaz, S.M., Jolai, F., Anaraki, A.H.Z.. A hybrid imperialist competitive algorithm for minimizing makespan in a multi-processor open shop. Applied Mathematical Modelling 2013;37(23):9603–9616.

[94] Enayatifar, R., Sadaei, H.J., Abdullah, A.H., Gani, A.. Imperialist competitive algorithm combined with refined high-order weighted fuzzy time series (rhwftsica) for short term load forecasting. Energy Conversion and Management 2013;76:1104–1116.

[95] Venske, S.M.S., Goncalves, R.A., Delgado, M.R.. Ademo/d: Adaptive differential evolution for multiobjective problems. In: Proceedings of the 2012 Brazilian Symposium on Neural Networks. SBRN '12; IEEE Computer Society; 2012, p. 226–231.

[96] de Melo, V.V., Carosio, G.L.. Investigating multi-view differential evolution for solving constrained engineering design problems. Expert Systems with Applications 2013;40(9):3370–3377.

[97] Mohamed, A.W., Sabry, H.Z., Abd-Elaziz, T.. Real parameter optimization by an effective differential evolution algorithm. Egyptian Informatics Journal 2013;14(1):37–53.

[98] Zhu, W., Tang, Y., an Fang, J., Zhang, W.. Adaptive population tuning scheme for differential evolution. Information Sciences 2013;223:164–191.

[99] Afanasyev, A., Moiseenko, I., Zhang, L.. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005; NDN; 2012. URL `named-data.net/techreports.html`.

[100] Dfn-verein: Dfn-noc (network operation center). 2013. Retrieved Jun. 2013; URL `dfn.de/dienstleistungen/dfninternet/noc`.

[101] Heckmann, O.. The competitive Internet service provider: network architecture, interconnection, traffic engineering and network design. Wiley series in communications networking & distributed systems; J. Wiley; 2006. URL `books.google.es/books?id=DyEfAQAAIAAJ`.