

Real-time transmission switching with neural networks

Bugaje, Al Amin B.; Cremer, Jochen L.; Strbac, Goran

DOI

[10.1049/gtd2.12698](https://doi.org/10.1049/gtd2.12698)

Publication date

2022

Document Version

Final published version

Published in

IET Generation, Transmission and Distribution

Citation (APA)

Bugaje, A. A. B., Cremer, J. L., & Strbac, G. (2022). Real-time transmission switching with neural networks. *IET Generation, Transmission and Distribution*, 17(3), 696-705. <https://doi.org/10.1049/gtd2.12698>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright



Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

ORIGINAL RESEARCH

Real-time transmission switching with neural networks

 Al-Amin B. Bugaje¹  | Jochen L. Cremer²  | Goran Strbac¹
¹Department of Electrical and Electronic Engineering, Imperial College London, London, UK

²Department of Electrical Sustainable Energy, TU Delft, Delft, The Netherlands
Correspondence
 Jochen L. Cremer, Department of Electrical Sustainable Energy, TU Delft, Delft, 2628 CD, The Netherlands.
Email: j.l.cremer@tudelft.nl
Funding information

Technische Universiteit Delft, Grant/Award Number: AI Labs Programme; Nigerian National Petroleum Corporation, Grant/Award Number: scholarship; Engineering and Physical Sciences Research Council, Grant/Award Number: EP/R045518/1

Abstract

The classical formulation of the transmission switching problem as a mixed-integer problem is intractable for large systems in real-time control settings. Several heuristics have been proposed in the past to speed up the computation time, which only limits the number of switchable lines. In this paper, a real-time switching heuristic based on neural networks that provides almost instantaneous switching actions, are presented. The findings are shown on case studies of the IEEE 118-bus test system, and the results show that the proposed heuristic is robust to out of distribution data. Additionally, the proposed heuristic has significant computational savings while all other performance metrics like accuracy are similar to state-of-the-art machine learning methods proposed for transmission switching.

1 | INTRODUCTION

Transmission switching (TS) involves the opening or closing of circuits or substations in a transmission network and has been used as a control mechanism by power system operators (SO) to improve voltage profiles and manage congestion on the electric network [1, 2]. Although power networks are planned with redundancies to handle multiple contingencies and deal with the uncertainty of future operating conditions (OCs), in real-time operation, SOs could use TS to efficiently operate the network infrastructure according to the loading conditions and generator costs. From a reliability perspective, circuits can be switched off to improve cost in normal operations and brought back during contingencies [1, 3]. Otherwise, TS happens while considering possible contingencies. Further work by Hedman [4] showed that TS does not inherently deteriorate reliability in the event of contingencies and Morsy [5] investigated the use of post contingency bus bar splitting to improve N-1 security after network reconfiguration. Importantly the short-term operations savings can be up to 25% of dispatch costs on the IEEE 118-bus system [1]. Other use cases of TS in power systems operations include reducing operating costs in a security-constrained unit commitment formulation that considers circuit breaker reliability

[6] and boosting resilience of power grids to extreme weather events [7].

Fisher's flagship paper [1] formulated the TS problem as a mixed-integer problem (MIP) that considers a DC optimal power flow (DCOPF) with binary variables tracking the state of lines as on or off. The formulation adopts the big-M method that constrains the line flows on switched-off lines to zero. As the SO schedules new dispatches every 5 min for the power system [8], solving for this optimised network topology and verifying as AC feasible ought to be done within that time frame. However, the computational burden of solving even the DC formulation of the TS problem in real-time prevents the adoption of TS in the control room as it is an NP-hard problem [9][10]. To quantify the magnitude of the search space, the IEEE 118-bus test case, which is small compared to real-world power systems, has 186 lines and thus 2^{186} switching possibilities. Exploring such a search space is intractable with modern-day computing power.

Heuristic approaches based on greedy local search using sensitivity-based algorithms [11, 12], constraint identification techniques [13], and power transfer distribution factors (PTDFs) [14], have been proposed to improve the computation speed. The sensitivity-based heuristics involve reducing the

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *IET Generation, Transmission & Distribution* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

large search space of the original MIP problem into a computationally tractable problem by ranking candidate switching actions. This ranking allows the easy exploration of possible ‘relevant’ line switches with the computationally inexpensive solution of multiple DCOPF problems. In refs. [11] and [12] the dual problem of the DCOPF informs how the authors rank the switchable lines based on a line’s tendency to improve the baseline dispatch cost. The baseline cost considers a topology with all lines as closed. A greedy local search of such a sensitivity-based ranking involves solving all possible DCOPFs for every possible line switching action [15, 8]. High performance computing (as in ref. [16]) and priority listing (as in refs. [11] and [17]) may be used to reduce the computation burden. The main drawback of these heuristics is their scalability to large networks, as they only provide a limitation on the number of switching options and then greedily search the reduced solution space. Granted, very few lines result in the largest cost reductions [1, 18]. An alternative sensitivity-based heuristic involves active constraint identification proposed by Crozier [13] to determine the need for TS in a congestion setting. The approach ranks the sensitivity of constraints on the dispatch cost when there is a mismatch between the DCOPF with economic dispatch (ED) and iteratively eliminates the constraints while solving a series of DCOPF problems. A different approach to the sensitivity-based heuristics avoids solving multiple DCOPFs and instead computes the power transfer distribution factors (PTDFs) as done in ref. [19]. However, these heuristics only work for a limited number of line switches and introduce sub-optimality resulting in higher normal operating costs than necessary.

Beyond numerous efforts to improve the solution quality of TS solutions, for instance in ref. [20] where an approximate model of the TS problem is proposed to speed up the computation time and improve solution quality was proposed, recently, the authors in ref. [21] propose an asynchronous algorithmic design that exploits domain-specific knowledge and heuristics in parallel to speed up the full NP-hard TS problem. New challenges that arise in uncertainty in variable renewable energy production [22] and in markets [23] have spurred further research into TS. For instance, the authors [22] consider a distributionally robust optimisation framework that considers uncertainty in power outputs of renewable energy sources and thereby aim to reduce the curtailment of renewable energy. The authors [23] formulate the day ahead and real-time zonal electricity market with TS as an adaptive robust optimisation mixed integer problem and present a new approach to solve the adversarial min-max problem which follows an interdiction game. However, overcoming the computational requirements to implement these methods in real-time settings remains a challenge.

Machine learning (ML) is promising to outperform heuristics in power system applications or make completely new applications feasible for the first time. For example, ML can simplify power system reliability studies [24] so they can run in real-time while avoiding sub-optimality otherwise introduced by heuristics. There, when selecting a supervised ML model for real-time reliability studies [25], the topological configuration can result in discrete changes in the underlying data distributions

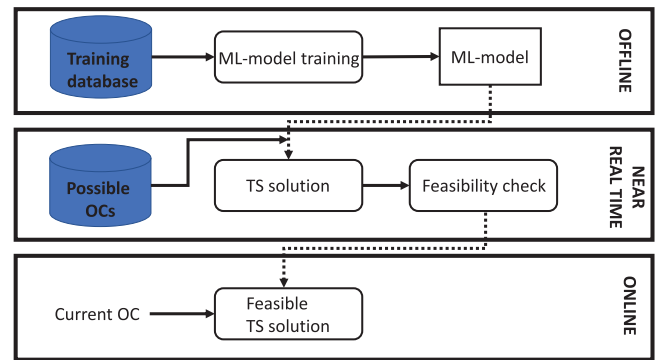


FIGURE 1 ML workflow for real-time topology change

that challenge the learned ML models [26]. Hence, exploring these discrete topological changes is an alternative that is then trained to a ML model through reinforcement rather than supervision [21]. Such explorations can enhance the operator’s experience and heuristics that would otherwise never use the explored actions.

To use supervised ML within the optimal transmission switching (OTS) framework is a new proposal to improve solution times of the OTS problem, as the canonical MIP formulation introduced by Fisher [1] is intractable in practical settings. A major advantage of the ML approach is the reduction in computation time required to select a candidate topology, which makes it a suitable approach to applying in real-time by the SO. The proposed ML workflow for TS is shown in Figure 1. There, the ML model is trained offline on historical operating data, and near-real time, proposes feasible topological configurations for possible operating conditions. After the physical feasibility of the proposed topology is confirmed, the proposed topology is configured in real-time. Previously, the authors in ref. [15] use ML to predict a list of high priority candidate line switches. There, ML models including a decision tree, a k -nearest neighbour (KNN), and a feed-forward multilayer perceptron artificial neural network (ANN) identify sets of suitable lines for TS. Moving the research further, Johnson [8] proposed a KNN heuristic to explicitly learn k -best candidate topologies for different load profiles, in similar fashion as the authors in ref. [27] that proposed two data-driven big-M bound strengthening methods (k -shortest path and k -nearest neighbour) that account for network topology, load demand and dispatch costs in the formulation. There, a classifier learns the relationship between operating conditions and their optimal topologies and proposes k nearest candidate topologies to evaluate the best topology for a new load profile. The results show that the KNN heuristic has negligible computational bottlenecks and can provide significant cost savings on dispatch costs. However, the KNN model faces scalability issues related to k -neighbours in higher dimensions [28]. Additionally, the robustness of the KNN heuristic to out of distribution data is in question.

Neural networks have previously been used in power system classification problems. [29] reviewed fault detection methodologies in power systems, and ANNs are suitable non-parametric models for the fault detection classification problem.

Due to the non-linear activation functions of ANNs, they can capture non-linear dependencies of complex dynamic systems as power systems [15]. The perceptron-layered network has been a popular ANN architecture for classification problems in power systems. Particularly, the 3-layered radial basis function neural networks (RBFNN) architecture is robust to inputs not in the training data [30]. This robustness makes it a favourite for power systems classification problems such as fault detection. RBFNNs [31] have been shown to perform well in terms of accuracy, interpretability, training data and time over other feed-forward architectures like multilayer perceptron and probabilistic neural networks in other fields as well.

The contributions in this paper are as follows:

1. We propose an RBFNN heuristic that is robust to noisy data to predict the TS solution in real-time settings considering \mathcal{J} maximum line switches. The proposed RBFNN architecture is a modification of the standard RBFNN architecture that includes K output layers instead of the original one layer to represent the transmission lines in a power system, and introduces a sigmoid activation function after each output layer to guarantee that all outputs lie in the range of $[0,1]$ to represent the status of the transmission lines. The proposed heuristic provides significant computational savings with similar accuracy performance as state-of-the-art ML heuristics for TS while outperforming line selection heuristics.
2. We propose a modified optimisation formulation of the TS problem to speed up the offline generation of training data by explicitly considering the switching action in the objective function.

The rest of the paper is structured as follows: Section 2 introduces the proposed RBFNN heuristic and describes the RBFNN architecture. We also briefly describe the KNN and greedy search heuristics we consider for comparison. Section 3 presents a case study that compares the performance of the proposed RBFNN heuristic with KNN and greedy-search heuristics relative to a baseline best-known solution obtained using Gurobi-based heuristics. Section 4 concludes the paper with an outlook for future work.

2 | HEURISTIC APPROACHES

This section introduces the proposed RBFNN heuristic and other state-of-the-art heuristics in the literature. We also present the computational complexity of the heuristic approaches we consider.

ML-based heuristics assume the availability of solved TS instances Ω^S to learn the mapping between input features (loading conditions) and output vectors (topology). Generating this set of solved TS instances Ω^S often requires solving a relaxed MIP problem. This paper adopts the MIP formulation in Fisher [1] slightly modified to penalise the objective function. The formulation considers a single period economic dispatch problem with binary variables to track switching decisions,

where ($z_l = 1$) and ($z_l = 0$) indicate available and unavailable lines, respectively.

The proposed modified objective function in Equation (1) explicitly considers the switching action alongside minimising the generation costs $C_g P_g$, thereby improving the solution time of the MIP problem. This explicit consideration of switching actions is the minimisation of the Euclidean distance $w_l(z_l^* - \alpha_l)^2$ between the binary variables z_l and the base topology that considers all lines as available $\alpha_l^{|\Omega|} \in \{1\}^{|\Omega|}$, where w_l is a weighting parameter. The physical constraints that ensure the capacity of power systems are met are in Equations (2)–(5) and the big-M method in Equations (6) and (7) ensures that when a line is switched off ($z_l = 0$), the line flow constraints on other lines that share same bus connections remain active. \mathcal{J} in Equation (8) sets an upper bound for the number of transmission lines that can be switched at all times.

All ML-based heuristics consider a set of TS solved instances Ω^S generated using the formulation in Equations (1)–(9).

$$\underset{P_g, z_l}{\text{minimize}} \quad \sum_g C_g P_g + \sum_l w_l (z_l - \alpha_l)^2 \quad (1)$$

subject to

$$\theta_n^{\min} \leq \theta_n \leq \theta_n^{\max}, \forall n \quad (2)$$

$$P_g^{\min} \leq P_g \leq P_g^{\max}, \forall g \quad (3)$$

$$P_l^{\min} z_l \leq P_l \leq P_l^{\max} z_l, \forall l \quad (4)$$

$$-\sum_l P_l - \sum_g P_g - \sum_d P_d = 0, \forall n \quad (5)$$

$$B_l(\theta_n - \theta_m) - P_l + (1 - z_l)M \geq 0, \forall l \quad (6)$$

$$B_l(\theta_n - \theta_m) - P_l - (1 - z_l)M \leq 0, \forall l \quad (7)$$

$$\sum_l (1 - z_l) \leq \mathcal{J} \quad (8)$$

$$z_l \in \{0, 1\}, \alpha_l \in \{1\} \quad (9)$$

In the rest of this section, we present the proposed RBFNN heuristics and describe other heuristics we consider for comparison as follows.

2.1 | Proposed RBFNN heuristic

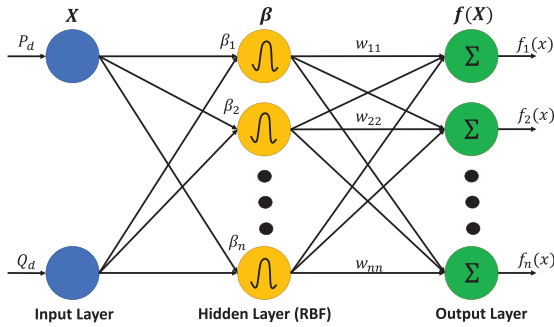
The proposed RBFNN heuristic is presented in Algorithm 1. The heuristic assumes the availability of many solved instances of the TS problem $\Omega^S = \{\{X_1, Y_1\}, \{X_2, Y_2\}, \dots, \{X_r, Y_r\}\}$ that covers a range of possible operating conditions, where X_r

ALGORITHM 1 Proposed RBFNN heuristic for real-time TS

Define set of solved TS instances
 $\Omega^S = \{\{X_1, Y_1\}, \{X_2, Y_2\}, \dots, \{X_r, Y_r\}\}$
 Train RBFNN using Ω^S
 To solve a new instance X_{r+1} , use X_{r+1} as input into
 RBFNN to get Y_{r+1}
 Solve:

$$\begin{aligned} & \underset{Y_{r+1}^*}{\text{minimize}} \sum_{i=1}^{|\Omega|} (Y_{r+1}^* - Y_{r+1}^i)^2 \\ & \text{s.t Eq. (2)-(9) are satisfied} \\ & \text{where } Y_{r+1}^i \in \{0, 1\}, \end{aligned}$$

$Y_{r+1} \leftarrow Y_{r+1}^*$
 return Y_{r+1}

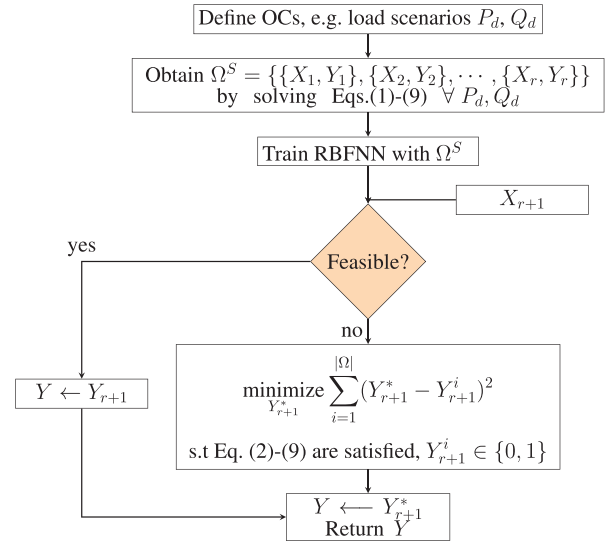
**FIGURE 2** An example RBFNN architecture

represents the loading condition and $f(X_r) = Y_r \in \{0, 1\}^{|\Omega|}$ are optimised topologies. Then, in real-time, the RBFNN considers active load profiles X_{r+1} as inputs and outputs a topology Y_{r+1} that can minimise the dispatch cost considering TS. An example architecture of the RBFNN (presented in Figure 2) is composed of three layers, the input layer, a non-linear hidden radial basis function (RBF) layer, and an output layer. The activation function of the non-linear hidden layer of the radial basis function neural network (RBFNN) architecture maps the input vector X

$$\Phi(X) = e^{-\beta \|x - \mu\|^2} \quad (10)$$

to a Gaussian function such that the feature space spans a set of Gaussian neural nodes [33]. The node centers μ are initialised with a K-means clustering algorithm, and the output of each node depends on $\|x - \mu\|$, such that similar inputs X in the Euclidean space generate similar outputs. β is a hyper-parameter of the hidden-layer to be tuned. The activation function of the output layer is a linear transformation of the weighted sum of all the outputs of the non-linear hidden layer w_{ij} . The training of the RBFNN with input vector X , hidden layer weights W , and output vector Y uses the least squares linear regression (LSLR) to obtain the weights

$$W = (X^T X)^{-1} X^T Y \quad (11)$$

**FIGURE 3** Flowchart of the proposed method showing all parameters

The exponential activation function allows the RBFNN to provide confidence intervals of predictions which is an important feature as SOs prefer to be in the decision loop when using ML-based approaches [32]. Similar RBFNN architectures are present in the literature, for example in refs. [33] and [34]. The key difference here is the consideration of K output layers instead of the original one layer to represent the transmission lines, and introduces a sigmoid activation function after each output layer to guarantee that all outputs lie in the range of $[0, 1]$.

Finally, in real-time, the RBFNN model accepts as input the current load profiles X_{r+1} and instantaneously outputs a TS solution Y_{r+1} . As it is possible that Y_{r+1} is not a feasible solution, the algorithm solves the optimisation

$$\underset{Y_{r+1}^*}{\text{minimize}} \sum_{i=1}^{|\Omega|} (Y_{r+1}^* - Y_{r+1}^i)^2 \quad (12)$$

while satisfying the constraints in Equations (2)–(7). Solving this optimisation returns the nearest feasible topology Y_{r+1}^* in the event that the TS solution obtained Y_{r+1} is not feasible. A detailed flowchart of the proposed method is shown in Figure 3 showing all parameters. The proposed method initialises by considering credible OCs, for example candidate load scenarios, P_i, Q_i , which serve as features X_r . Then, for each OC, X_r , the proposed method solves Equations (1)–(9) to obtain the optimal TS solution Y_r , given J maximum available switching actions. The set Ω^S comprising of possible OCs as features X_r with their respective optimal topologies Y_r as outputs becomes the training data of the RBFNN. Then, in real-time operation, for a given OC X_{r+1} , the SO can consider this trained RBFNN model as a heuristic to obtain a TS solution Y_{r+1} to relieve congestion and reduce dispatch cost. The proposed heuristic performs a quick feasibility test on the new topology Y_{r+1} , and returns the nearest feasible topology Y_{r+1}^* in the event of an infeasible solution Y_{r+1} .

2.2 | KNN heuristic

The KNN heuristic is based on Johnson [7] where a mapping exists between vectors of OCs X and the known DC optimal switching instances $f(X)$ resulting in a set Ω^S of TS solved instances. The KNN heuristic trains on the training data Ω^S , and in real-time chooses k instances that are similar to a candidate OC X based on the Euclidean distance or l_∞ -norm. Subsequently, k -DCOPFs are solved and the TS instance with the lowest dispatch cost is returned as the selected switching action.

2.3 | Sensitivity-based greedy search heuristic

We consider the heuristic proposed in Fuller [11] to rank the lines according to the sensitivity parameter of the dual of the DCOPF (nodal price). Subsequently, the algorithm iterates over half the ranked lines in a sequential manner as done by Yang [15] termed the line enumeration algorithm. The algorithm starts with a baseline cost considering a topology with all lines in service. At the i th iteration, the algorithm then solves $|\Omega| - i$ DCOPF problems by opening single lines according to the line ranking and permanently switching off the line that results in the most cost improvement. Starting with $i = 0$, the algorithm continues until there is no cost improvement or the upper limit of switchable lines $\sum_j (1 - z_j) = J$ is satisfied.

2.4 | Gurobic heuristic

The Gurobi heuristic solves for the optimal TS problem described in Equations (1)–(9). We consider this formulation as a heuristic as there is a limit to the upper-bound of switched lines $\sum_j (1 - z_j) \leq J$. This heuristic serves as the yardstick to compare the different heuristics.

2.5 | Computational complexity

The computational complexity of the different heuristics differ depending on whether the comparison is offline or online. In the offline setting, the ML approaches generally require a significant amount of time that increases relative to the size of the test network, as computationally expensive MIPs need to be solved to curate the set Ω^S . Training the RBFNN varies with the number of neurons in the architecture and the number of epochs, but the simple architecture of the RBFNN allows for a linear training time. The offline training time for the KNN heuristic is dependent on the k -neighbours. The offline computation for the sensitivity-based greedy local search heuristic is trivial as it requires only solving a DCOPF.

In the online setting where there is a huge constraint in time, an exhaustive search for the greedy search heuristic requires $\mathcal{O}(|\Omega|^2)$ DCOPF solves and thus scales poorly in large systems. The KNN heuristic depends on the k DCOPFs required to be

solved. The RBFNN heuristic provides almost instantaneous solutions that requires at most a single DCOPF solve to verify feasibility. Thus, we expect that the proposed RBFNN heuristic will provide significant computational improvements in the online setting even with the feasibility verification step.

3 | CASE STUDY

In this section, we carry out case studies to compare the proposed RBFNN heuristic with the KNN, sensitivity-based greedy search, and Gurobi-based heuristics. The first study investigates the performance of the heuristics to out of distribution data as a measure of robustness. In the second and third study, we compare the real-time performance of the heuristics in respect to computational time and cost savings, respectively. The fourth case study highlights the offline computational improvements of the modified MIP formulation. Finally, a discussion section concludes the case study section with key findings.

3.1 | Test system and assumption

The case studies use the modified IEEE 118-bus test case in Blumsack [35]. The load profiles are generated via Monte Carlo sampling. We sampled the active loads from a multivariate Gaussian distribution and assume the correlation between loads to follow Pearson's correlation with a correlation coefficient of 0.75. The distribution was then converted to a marginal Kumaraswamy (1.6,2.8) distribution using inverse transformation. We consider $\pm 10\%$ variation in the load distribution for the training data-set. We then run the optimisation in Equations (1)–(9) for each load profile with an upper-bound on switching actions $J \leq 5$. The choice of $J \leq 5$ is from the literature, as only a few lines result in the largest cost savings [1, 3] and the average congestion cost savings plateau around $J = 5$ lines for the IEEE 118 bus test system [11]. However, we also extend the maximum number of switchable lines to $J \leq 10$ in a case study to show that the performance of the proposed algorithm is scalable. The training data is then a feature space comprising 1000 active load profiles $X_r \in \{P_j\}$ and labels that correspond to a binary sequence $Y_r \in \{0, 1\}^{|\Omega|}$, where $Y_r^i = 0$ and $Y_r^i = 1$ represents the i th line as absent and present, respectively. In studying the real-time computational performance, all previously introduced heuristics are tested on 200 active load profiles with $\pm 35\%$ variation in the load distribution. Unless otherwise stated, all studies consider the $\pm 35\%$ variation in the testing data.

All optimisation problems were implemented using the package Pyomo 5.6.8 [36] in Python 3.7.4 and solved using Gurobi 9.5.0 [37]. The RBFNN is implemented using Keras package 2.8.0, with root mean squared propagation (RMSprop) as the optimiser and mean squared error as the loss function. The activation function between the input and hidden layer, and between the hidden and output layer is linear. We also include

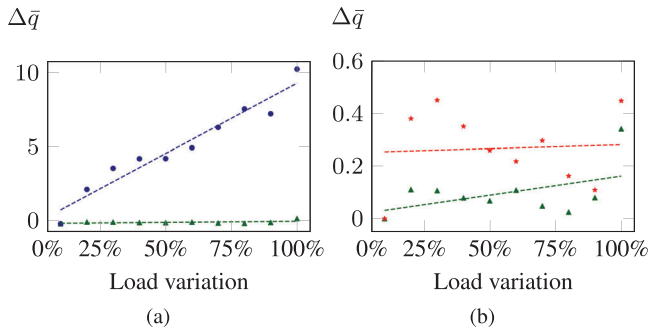


FIGURE 4 Change in average dispatch cost ($\Delta\bar{q}$) with variation in load distribution for the proposed RBFNN (\blacktriangle) and (a) KNN (\bullet) (b) a feed-forward ANN (\star) heuristics

a sigmoid activation function after the output layer such that all outputs lie in the range $[0,1]$. The β parameters of the hidden layer were 0.001, and the training considered an epoch length of 200 and a batch size of 5.

3.2 | Performance of heuristics on out of distribution data

This case study investigates the performance of the proposed RBFNN heuristic to out of distribution data. This study compares the proposed RBFNN heuristic against the KNN heuristic and a feed-forward perceptron ANN. The feed-forward ANN follows Algorithm (1) similar to the proposed RBFNN with the difference only in the model architecture of five as opposed to three layers. All the heuristics were trained considering a $\pm 10\%$ variation in the load distribution as described in Section 3.1. The baseline performance is considered using data not in the training set but following a similar $\pm 10\%$ variation. Subsequently, we compare the relative change in performance of the heuristics on test instances that differ from the $\pm 10\%$ variation.

The results are shown in Figure 4(a), which presents a scatterplot that compares the change in average relative dispatch cost with different variations (10%, 20%, 30%, ..., 100%) in load distribution for the proposed RBFNN, KNN and feed-forward ANN heuristics. The average relative change in dispatch costs considers a baseline where the training and testing data come from the same distribution of $\pm 10\%$ variation. Subsequently we test the heuristics for each variation in load distribution and compute the change ($\Delta\bar{q} = \frac{\bar{q}-q}{q}$) from the baseline relative cost

q for 200 test cases in $\Omega^{s''}$, where $\bar{q} = \frac{\sum_{i=1}^{|\Omega^{s''}|} \hat{q}-q}{|\Omega^{s''}|}$ is the average relative cost for an heuristic.

The results in Figure 4(a) show a minimum and maximum change in relative costs $\Delta\bar{q}$ of 2.5% and 34.6%, respectively, corresponding to the $\pm 20\%$ and $\pm 100\%$ load variation for the proposed RBFNN heuristic. The linear regression line is horizontal, which denotes little to no change and thus robustness to out of distribution data. In contrast, the KNN heuristic has a minimum and maximum change $\Delta\bar{q}$ of 224% and 1000%,

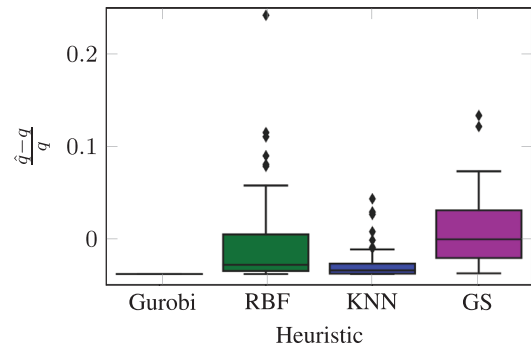


FIGURE 5 The relative dispatch cost to best known solution ($\frac{\hat{q}-q}{q}$) for 200 test instances using different heuristics

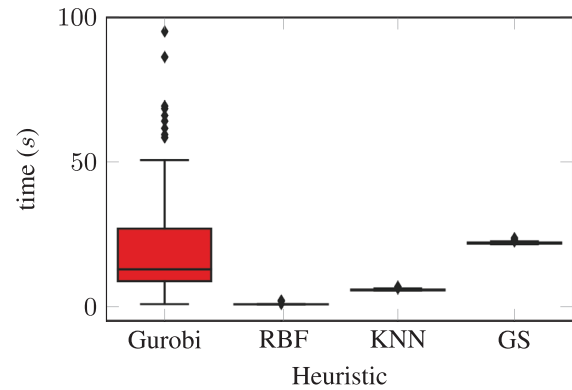


FIGURE 6 Computational time to obtain TS solutions for 200 test instances using different heuristics

respectively, corresponding to the $\pm 20\%$ and $\pm 100\%$ load variation. Additionally, the linear regression line makes an $\approx 45^\circ$ with the horizontal axis, which suggests that the heuristic performance changes with different load variations. A further comparison between the proposed RBFNN heuristic with a feed-forward ANN heuristic for similar changes in load distribution in Figure 4(b) shows that the proposed RBFNN has a similar regression line as a feed-forward ANN. There, the feed-forward ANN has a minimum and maximum change of 10.9% and 45.2%, respectively. While both the proposed RBFNN and the feed-forward ANN have an average relative cost $\frac{\bar{q}-q}{q}$ of $\leq 5\%$, the average change $\Delta\bar{q}$ over the different variations (10%, 20%, ..., 100%) for the proposed RBFNN is 9.7% while that of the feed-forward ANN is 26.7%. These results support the robustness of the proposed RBFNN heuristic.

3.3 | Real-time computational performance

This case study shows the computational results of the heuristics introduced in Section 2 in terms of computational time. While the ML-based heuristics training was on data from the $\pm 10\%$ variation, the testing data had a $\pm 35\%$ variation. The box plots in Figure 6 and the data in Table 1 summarise the computational time results. The box plot indicates the median value as

TABLE 1 The relative cost and computation time for different heuristics on 200 TS test instances considering $\mathcal{J} \leq 5$ switching actions

Heuristic	Relative cost ($\frac{\hat{q}-q}{q}$)	Time
Gurobi	0.00 \pm 0.00	31.31 \pm 61.38s
RBFNN	0.03 \pm 0.03	0.57 \pm 0.21s
KNN	0.01 \pm 0.02	6.25 \pm 0.25s
Greedy search	0.04 \pm 0.03	21.93 \pm 0.32s

TABLE 2 The relative cost and computation time for different heuristics on 200 TS test instances considering $\mathcal{J} \leq 10$ switching actions

Heuristic	Relative cost ($\frac{\hat{q}-q}{q}$)	Time
Gurobi	0.00 \pm 0.00	31.31 \pm 61.38s
RBFNN	0.03 \pm 0.03	0.57 \pm 0.21s
KNN	0.01 \pm 0.02	6.25 \pm 0.25s
Greedy search	0.04 \pm 0.03	21.93 \pm 0.32s

the middle line in the box, the first and third quartiles as the top and bottom lines in the box, respectively, and the outliers as individual points.

Concretely, Figure 6 presents the distribution of the computational time it takes to solve the TS problem for 200 test instances. The proposed RBFNN heuristic has a median value of 1.2s, while the Gurobi, KNN and greedy search heuristics have median values of 13s, 6s, 22s, respectively. This result is supported by the data in Table 1 that shows the mean and standard deviation of the computation time for the different heuristics. There, the proposed RBFNN heuristic outputs a TS solution on average in 0.57s, while the KNN, Greedy Search and Gurobi heuristics average 6.25s, 21.93s, and 31.31s respectively. We consider a computational upper bound of 50s to the solver for the Gurobi heuristic, otherwise, average computational times are as long as 300s [8]. This massive computational saving makes the proposed RBFNN a suitable approach for real-time applications. These results also match our computational expectations described in Section 2.5. We extend the maximum allowable switchable lines to $\mathcal{J} \leq 10$ to investigate the scalability of the proposed approach, which the results in Table 2 show similar computational performance for the proposed RBFNN heuristic and the KNN heuristic. This consistent computational performance is a key feature of ML methods. On the other hand, there are notable increases in the computational time for the greedy search and Gurobi heuristics. For $\mathcal{J} \leq 10$, we consider a computational upper bound of 120s to the solver for the Gurobi heuristic, otherwise, average computational times are as long as 300s [8].

3.4 | Cost performance

This case study compares the savings in the dispatch cost from using the heuristics introduced in Section 2. Figure 5 com-

TABLE 3 Handling infeasible TS of the proposed RBFNN heuristic for $\mathcal{J} \leq 5$

Heuristic	Infeasible TS	Time	Relative cost ($\frac{\hat{q}-q}{q}$)
RBFNN	4/200	1.32 \pm 0.05s	0.08 \pm 0.10
Gurobi	0/200	7.25 \pm 2.62s	0.00 \pm 0.00

pare the relative cost savings using the different heuristics for 200 test instances. The relative costs are calculated according to the best known dispatch cost (calculated using the Gurobi heuristic) as $\frac{\hat{q}-q}{q}$, where \hat{q} represents the dispatch cost obtained using an heuristic for TS, and q is the best known cost. Concretely, the figure shows that the proposed RBFNN heuristic with a median relative cost of 0.8%, outperforms the greedy search heuristics at 3.3%, and is similar to the KNN heuristic at 0.3% relative cost. This result is consistent with the data in Table 1 showing the mean and standard deviation of the relative costs for the different heuristics. The results in Table 1 show that the proposed RBFNN heuristic is on average 3% close to the best known TS solution. This result suggests that the proposed RBFNN heuristic can propose good TS solutions. We extend the maximum allowable switchable lines to $\mathcal{J} \leq 10$ to investigate the scalability of the proposed approach, which the result in Table 2 shows similar cost performance for the proposed RBFNN heuristic being within 4% of the best-known solution.

3.5 | Guaranteeing feasible topology near real-time

This case study investigates the handling of infeasible TS solutions by the proposed RBFNN heuristic near real-time. Notably, the crucial comparison is between the proposed RBFNN and the KNN heuristics, as the scalability of the greedy search heuristic is poor due to $\mathcal{O}(|\Omega|^2)$ DCOPF computations required. While the KNN offers a marginally better relative cost value, the value of k determines the number of DCOPF problems to solve and thus worsens the computational time. Sometimes, however, the initial predicted solution by the RBFNN is infeasible and in such cases the feasibility is ensured by solving Equation (12) that returns the nearest feasible topology. In this case study, these infeasible solutions appeared in 4 out of 200 test instances (2%). For these four infeasible TS instances, the proposed heuristic took an average of 1.32s to obtain the nearest feasible topology and the obtained result was within 8% of the best-known solution considering $\mathcal{J} \leq 5$ line switches, as opposed to an average of 7.25s required to solve the same MIP formulation with a similar relaxation. These results are summarised in Table 3 and importantly the proposed RBFNN solves for a feasible TS topology within 8% of the best-known solution while being more than 5X faster than solving Fisher's [1] MIP formulation.

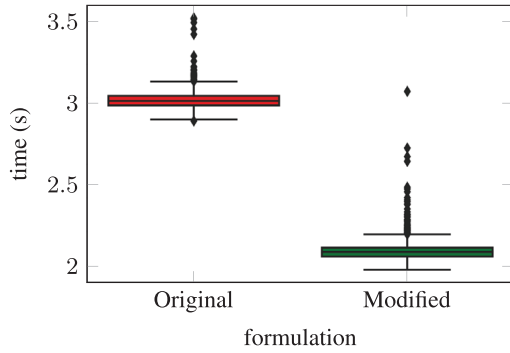


FIGURE 7 Solution time for different MIP formulations

3.6 | Offline computational improvements

In this case study, we investigated the effect of penalising the objective function on the solution times of the MIP. We considered a thousand solution instances of the MIP problem for a given load profile with $\mathcal{J} \leq 3$. The original formulation does not consider the penalty term $\sum_l w_l (\mathcal{Z}_l^* - \alpha_l)^2$ in Equation (1), while the modified formulation is exactly as presented in Section 2. The results are presented as box-plots of solution times in seconds in Figure 7. Concretely, the results show an average improvement of 0.90s for the modified formulation, which is 43% faster than the original MIP formulation. This percentage improvement does not extrapolate a linear relationship as the maximum number of number of line switches \mathcal{J} increase. However, as the number of crucial line switches are usually few, this result is still relevant.

3.7 | Key findings

Within ML, the RBFNN model has been shown to be robust to inputs not in the training data. In our studies, the proposed RBFNN TS heuristic showed also high accuracy performance across different data distributions, with a maximum deviation of 34.6% in contrast to the 1000% deviation of the KNN model. RBFNN models have been shown to perform well in terms of accuracy, interpretability, training data, and computation time. Similarly, RBFNN models show significant benefits of providing almost instantaneous TS options and thus are applicable in real-time settings by SOs. The main computational burden comes from generating the solved TS instances for training the RBFNN model which is done offline. However, this offline step is necessary for any ML workflow that require synthetic data to train a model. A downside of the proposed approach is that sometimes (in 2% cases) the RBFNN TS heuristics results in infeasible TS in realtime, however, there, we proposed an efficient optimisation TS formulation that mitigates the issue, resulting in near-real-time TS. On a positive note, the simplicity of the RBFNN makes training the model relatively fast and its architecture allows for more interpretability of the model. The proposed RBFNN has an error margin of 3% relative to the best known solution and pro-

vides the TS solution more than 10× faster than state of the art ML-based TS.

4 | CONCLUSION

This work proposed a RBFNN heuristic for real-time TS. The proposed heuristic outperforms heuristics in the literature such as KNN models and sensitivity-based greedy search algorithms in terms of significant savings in computation time. The proposed RBFNN heuristic is fairly robust to noisy data and performs as well as the state of art ML approach to TS. In situations where the proposed RBFNN outputs an infeasible topology, the proposed RBFNN solves for a feasible TS topology that is within 8% of the best-known solution while being more than 5× faster than solving a standard MIP. We test our approach on the IEEE 118-bus test case and limit the maximum number of switchable lines to $\mathcal{J} \leq 5$. As the issue of contingencies is of utmost importance to the SO, further work will investigate learning a model to propose TS actions that are N-1 secure. Finally, as ML becomes even more crucial in power systems dominated by converter interfaced generation [38], future work shall also consider the impact of uncertainties from variable energy sources like photovoltaic and wind turbines under different operation scenarios.

NOMENCLATURE

Indices

- d Index of demands
- g Index of generators
- l Index of transmission lines
- n Index of buses

Sets

- Ω Set of transmission lines
- Ω^S Set of transmission switching solution instances
- $\Omega^{S'}$ Set of transmission switching solution instances for testing
- $\Omega^{S''}$ Set of transmission switching solution instances for training

Parameters

- \bar{q} Average dispatch cost using an heuristic
- β Shape parameter of neural network activation function
- \hat{q} Dispatch cost using a TS heuristic
- \mathcal{J} Maximum number of switchable lines
- μ Position parameter of neural network activation function
- $\phi(X)$ Activation function of neural network hidden layer
- a_l Status of line l
- B_l Susceptance of line l
- C_g Linear cost of generator g
- k Number of nearest neighbours
- M Big-M value
- P_g^{\max} Maximum power for generator g
- P_g^{\min} Minimum power for generator g

P_d	Active power load
P_l^{\max}	Maximum flow on line l
P_l^{\min}	Minimum flow on line l
q	Best known dispatch cost from transmission switching
w_l	Weight of penalty function in optimisation
X	Input to neural network

Variables

θ_n	Bus angle variable
P_g	Generator variable
P_l	Transmission flow variable
W	Weights for neural network training
Y	Output of neural network
z_l	Binary variable defining line status as on or off

AUTHOR CONTRIBUTIONS

Al-Amin Bugaje: Conceptualization, investigation, methodology, visualization, writing - original draft. Jochen Cremer: Conceptualization, formal analysis, investigation, supervision, writing - review and editing. Goran Strbac (GE): Funding acquisition, project administration, resources, supervision, writing - review and editing.

ACKNOWLEDGEMENTS

This work was supported by a scholarship funded by the Nigerian National Petroleum Corporation, NNPC, the TU Delft AI Labs Programme, NL, and the research project IDLES, UK (EP/R045518/1).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Al-Amin B. Bugaje  <https://orcid.org/0000-0002-2993-5710>
Jochen L. Cremer  <https://orcid.org/0000-0001-9284-5083>

REFERENCES

- Fisher, E.B., O'Neill, R.P., Ferris, M.C.: Optimal transmission switching. *IEEE Trans. Power Syst.* 23(3), 1346–1355 (2008)
- Khanabadi, M., Ghasemi, H.: Transmission congestion management through optimal transmission switching. In: *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–5. IEEE, Piscataway, NJ (2011)
- Hedman, K.W., O'Neill, R.P., Fisher, E.B., et al.: Optimal transmission switching—sensitivity analysis and extensions. *IEEE Trans. Power Syst.* 23(3), 1469–1479 (2008)
- Hedman, K.W., O'Neill, R.P., Fisher, E.B., et al.: Optimal transmission switching with contingency analysis. *IEEE Trans. Power Syst.* 24(3), 1577–1586 (2009)
- Morsy, B., Hinneck, A., Pozo, D., et al.: Security constrained OPF utilizing substation reconfiguration and busbar splitting. *Electr. Power Syst. Res.* 212, 108507 (2022)
- Sheikh, M., Aghaei, J., Letafat, A., et al.: Security-constrained unit commitment problem with transmission switching reliability and dynamic thermal line rating. *IEEE Syst. J.* 13(4), 3933–3943 (2019)
- Hussain, T., Suryanarayanan, S., Hansen, T.M., et al.: A fast and scalable transmission switching algorithm for boosting resilience of electric grids impacted by extreme weather events. *IEEE Access* 10, 57893–57901 (2022)
- Johnson, E.S., Ahmed, S., Dey, S.S., et al.: A k-nearest neighbor heuristic for real-time DC optimal transmission switching. *arXiv:2003.10565* (2020)
- Lehmann, K., Grastien, A., Van Hentenryck, P.: The complexity of de-switching problems. *arXiv:1411.4369* (2014)
- Kocuk, B., Jeon, H., Dey, S.S., et al.: A cycle-based formulation and valid inequalities for DC power transmission problems with switching. *Oper. Res.* 64(4), 922–938 (2016)
- Fuller, J.D., Ramasra, R., Cha, A.: Fast heuristics for transmission line switching. *IEEE Trans. Power Syst.* 27(3), 1377–1386 (2012)
- Ruiz, P.A., Foster, J.M., Rudkevich, A., et al.: Tractable transmission topology control using sensitivity analysis. *IEEE Trans. Power Syst.* 27(3), 1550–1559 (2012)
- Crozier, C., Baker, K., Toomey, B.: Feasible region-based heuristics for optimal transmission switching. *Sustainable Energy Grids Networks* 30, 100628 (2022)
- Ruiz, P.A., Goldis, E., Rudkevich, A.M., et al.: Security-constrained transmission topology control milp formulation using sensitivity factors. *IEEE Trans. Power Syst.* 32(2), 1597–1605 (2016)
- Yang, Z., Oren, S.: Line selection and algorithm selection for transmission switching by machine learning methods. In: *Proceedings of the IEEE Milan PowerTech*, pp. 1–6. IEEE, Piscataway, NJ (2019)
- Papavasiliou, A., Oren, S.S., Yang, Z., et al.: An application of high performance computing to transmission switching. In: *Proceedings of the IREP Symposium Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid*, pp. 1–6. IEEE, Piscataway, NJ (2013)
- Soroush, M., Fuller, J.D.: Accuracies of optimal transmission switching heuristics based on DCOPF and ACOPF. *IEEE Trans. Power Syst.* 29(2), 924–932 (2013)
- Barrows, C., Blumsack, S., Bent, R.: Computationally efficient optimal transmission switching: Solution space reduction. In: *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–8. IEEE, Piscataway, NJ (2012)
- Ruiz, P.A., Rudkevich, A., Caramanis, M.C., et al.: Reduced MIP formulation for transmission topology control. In: *Proceedings of the 50th Annual Conf. on Communication, Control, and Computing*, pp. 1073–1079. IEEE, Piscataway, NJ (2012)
- Jabarnejad, M.: Approximate optimal transmission switching. *Electr. Power Syst. Res.* 161, 1–7 (2018)
- Hinneck, A., Pozo, D.: Optimal transmission switching: improving exact algorithms by parallel incumbent solution generation. *IEEE Trans. Power Syst.* (2022). <https://doi.org/10.1109/TPWRS.2022.3199114>
- Zhou, Y., Zhu, H., Hanasusanto, G.A.: Distributionally Robust Chance-Constrained Optimal Transmission Switching for Renewable Integration. *IEEE Trans. Sustainable Energy* (2022). <https://doi.org/10.1109/TSTE.2022.3203669>
- Lété, Q., Papavasiliou, A.: Impacts of transmission switching in zonal electricity markets-part I. *IEEE Trans. Power Syst.* 36(2), 902–913 (2020)
- Duchesne, L., Karangelos, E., Wehenkel, L.: Recent developments in machine learning for energy systems reliability management. *Proc. IEEE* 108(9), 1656–1676 (2020)
- Bugaje, A.-A. B., Cremer, J.L., Sun, M., et al.: Selecting decision trees for power system security assessment. *Energy AI* 6, 100110 (2021)
- Bellizio, F., Cremer, J.L., Strbac, G.: Machine-learned security assessment for changing system topologies. *Int. J. Electr. Power Energy Syst.* 134, 107380 (2022)
- Li, J., Dokka, T., Lulli, G., et al.: Data-driven Heuristics for DC optimal transmission switching problem. *arXiv:2110.08141* (2021)
- Kouiroukidis, N., Evangelidis, G.: The effects of dimensionality curse in high dimensional KNN search. In: *Proceedings of the 15th Panhellenic Conference on Informatics*, pp. 41–45. IEEE, Piscataway, NJ (2011)
- Aleem, S.A., Shahid, N., Naqvi, I.H.: Methodologies in power systems fault detection and diagnosis. *Energy Syst.* 6(1), 85–108 (2015)

30. Khalkho, A.M., Mohanta, D.K.: RBFNN assisted transient instability based risk assessment of cyber-physical power system. *Int. J. Electr. Power Energy Syst.* 137, 107787 (2022)
31. Lassoued, H., Ketata, R., Yacoub, S.: ECG decision support system based on feedforward neural networks. *Int. J. Smart Sens. Intell. Syst.* 11(1), 1–13 (2018)
32. Marot, A., Donnot, B., Chaouache, K., et al.: Learning to run a power network with trust. *Electr. Power Syst. Res.* 212, 108487 (2022)
33. Haykin, S., Network, N.: A comprehensive foundation. *Neural Networks* 2, 41 (2004)
34. Karami, A., Mohammadi, M.S.: Radial basis function neural network for power system load-flow. *Int. J. Electr. Power Energy Syst.* 30(1), 60–66 (2008)
35. Blumsack, S.: Network topologies and transmission investment under electric industry restructuring. Ph.D. Thesis, Carnegie Mellon University (2006)
36. Hart, W.E., Laird, C.D., Watson, J.-P., et al.: *Pyomo-Optimization Modeling in Python*. Springer, Cham (2017)
37. Gurobi: *Gurobi Optimizer Reference Manual*. Gurobi Optimization LLC, Houston, TX (2018)
38. Bellizio, F., Bugaje, A.-A.B., Cremer, J.L., Strbac, G.: Verifying machine learning conclusions for securing low inertia systems. *Sustainable Energy Grids Networks* 30, 100656 (2022)

How to cite this article: Bugaje, A.-A. B., Cremer, J.L., Strbac, G.: Real-time transmission switching with neural networks. *IET Gener. Transm. Distrib.* 1–10 (2022). <https://doi.org/10.1049/gtd2.12698>