Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 4, 2023, 627-638.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research Paper

Applying Twin-Hybrid Feature Selection Scheme on Transient Multi-Trajectory Data for Transient Stability Prediction

Seyed Alireza Bashiri Mosavi[1]* and Omid Khalaf Beigi[2]

1. Department of Electrical and Computer Engineering, Buein Zahra Technical University, Buein Zahra, Qazvin, Iran.
2. Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran.

Article Info

Article History:
Received 02 September 2023
Revised 12 October 2023
Accepted 31 October 2023

DOI:10.22044/jadm.2023.13545.2472

Keywords:
Hybrid Feature Selection Scheme, Relevant-discriminative Transient Features, Transient Stability Prediction.

Corresponding author: abashirimosavi@bzte.ac.ir (S. A. Bashiri Mosavi).

Abstract

A speedy and accurate transient stability assessment (TSA) is gained by employing efficient machine learning- and statistics-based (MLST) algorithms on transient nonlinear time series space. In the MLST's world, the feature selection process by forming compacted optimal transient feature space (COTFS) from raw high dimensional transient data can pave the way for high-performance TSA. Hence, designing a comprehensive feature selection scheme (FSS) that populates COTFS with the relevant-discriminative transient features (RDTFs) is an urgent need. This work aims to introduce twin hybrid FSS (THFSS) to select RDTFs from transient 28-variate time series data. Each fold of THFSS comprises filter-wrapper mechanisms. The conditional relevancy rate (CRR), based on mutual information (MI) and entropy calculations, are considered as filter method, and incremental wrapper subset selection (IWSS) and IWSS with replacement (IWSSr) formed by kernelized support vector machine (SVM) and twin SVM (TWSVM) are used as the wrapper ones. After exerting THFSS on transient univariates, RDTFs are entered into the cross-validation-based train-test procedure for evaluating their efficiency in TSA. The results manifested that THFSS-based RDTFs have a prediction accuracy of 98.87 % and a processing time of 102.653 milliseconds for TSA.

1. Introduction

Nowadays, continuous monitoring of dynamic features related to power system reliability (PSR) is tightly correlated to conducting data mining (DM) techniques [1, 2] on power raw high-dimensional transient space (RHDTS) retrieved by wide-area measurement systems (WAMS) for knowledge discovery. Regardless of the high precision DM-based predictive models (DMPMs) mounted on IT-based infrastructure for grid monitoring, these learning methods should be reformed based on the type of under-study phenomenon in PSR scope. One of the most significant branches of PSR analysis is transient stability proposed by Kundur *et al.* [3], known for its fast character [4]. In such circumstances, the power grid transient monitoring demands timely analysis from DMPMs for speedy control actions. Hence, the DMPM design should satisfy fast and accurate transient stability assessment (TSA). In the presence of RHDTS, achieving such a dual-purpose-based TSA would be impossible. This problem originates from two factors: 1) high dimensions in RHDTS, which cause high prediction time, and 2) populating RHDTS with the irrelevant-non discriminative transient features (IRNDTFs) that lead to low accuracy prediction. The best way that can solve the paradoxical challenges derived by RHDTS leading to low-performance TSA is feature selection [5]. The feature selection process, by discarding IRNDTFs from the RHDTS, forms the relevant-discriminative transient feature (RDTF) set as compacted optimal transient feature space (COTFS), which promises high-level information sharing between COTFS's members and target class. COTFS-based train-test procedure (CTTP) plugged into DMPM brings high-performance TSA from two aspects: 1) extracting important patterns via RDTFs causes high prediction accuracy of transient unseen cases, and 2) the fast execution in CTTP due to low dimensional space reduces the prediction time as the main component of the processing time of transient sample labeling. The above concerns remind the importance of designing a comprehensive feature selection scheme (FSS) coordinated as a joint study by data mining and electrical engineers for the high-performance TSA.

2. Related Works

A glance over FSS-based transient works reveals that COTFS has been formed by filter- or hybrid-oriented FSS. In the case of filter-based FSS, there are so many works that find optimal features via information theory-based approaches [6-9]. For example, mutual information (MI)-based FSS can be found in Li *et al.* [6] and Liu *et al.* [7] works, which selects the most relevant power and angle features for transient stability prediction (TSP). For induction motor analysis, Stief *et al.* [8] utilizes the ReliefF algorithm to eliminate redundant features. In the transfer capability calculation framework developed by Yan *et al.* [9], applying the fast correlation-based filter (FCBF) on high-dimensional data is considered to select optimal features for high-performance TSP. In the feature selection-based transient studies, a diverse range of hybrid FSS has been proposed by scholars [10-16]. For example, Ji *et al.* [10] selects the optimal transient features by Relief-support vector machine (SVM) as a filter-wrapper algorithm (FWA) for power system dynamic monitoring. As another hybrid-faced FSS, Chen *et al.* [11] considers coupling normalized MI and binary particle swarm optimization as FWA for high-performance TSP. The fuzzy imperialist competitive-based hybrid FSS with a point-and trajectory-oriented attitude in the feature selection process is introduced by Bashiri Mosavi *et al.* [12] to find discriminative transient features. Surviving optimal transient features by embedding kernelized fuzzy rough sets (KFRS) in binary Jaya-based FSS and memetic algorithm for forming hybrid FSS can be found in Li *et al.* [13] and Gu *et al.* [14] works, respectively. The cross-permutation-based quad-hybrid FSS (CPQHFSS) developed by Bashiri Mosavi *et al.* [15] presents polyhedral FWA for finding optimal transient features. Bashiri Mosavi *et al.* [16] offer trilateral nested filter-wrapper FSS, which is applied on transient excursions in a partial-oriented manner for selecting optimal-blurred transient features.

Regardless of the considerable structural difference between filter and hybrid FSS in feature selection, these schemes suffer from two problems, namely vertically singular strategy (VSS) and chained trajectory features (CTFs) case. The VSS refers to a non-cross-linked learning model in forming COTFS. Such a strategy which can be seen in works like [6-9], [10-12], [15], [16], paves no way for semi-optimal transient features (SOTFs). (The SOTF is a feature in which the filter- or wrapper-based index value (FWIV) is closest to the optimal feature-specific FWIV, and their presence next to optimal features may amplify the model's training power) to enter into COTFS. Hence, it is necessary to design a cross-linked hybrid FSS (CLHFSS) that makes the presence of SOTFs in the optimal features' neighborhood. The CTFs is related to how to feed data to the CLHFFS, and can negatively affect the feature selection process. For example, Li *et al.* [13] and Gu *et al.* [14] mount their proposed FSSs on a single transient trajectory derived by placing time series consecutively. This issue can exacerbate the ignoring of optimal transient features (primarily) and SOTFs (in second-degree) per trajectory. Entering unchained trajectory features into the CLHFSS (each trajectory feature is entered into the feature selection process independently) leads to selecting optimal and semi-optimal features per trajectory.

This paper's contributions to overcoming the two major weaknesses of the past feature selection algorithms are listed below:

- A new scheme named the twin hybrid FSS (THFSS) is proposed to select RDTFs for speedy-accurate TSP. Each fold of the THFSS contains filter-wrapper scenarios.
- The UTFs view to find the optimal-blurred features from transient 28-variate time series data (T28VTD) is followed by THFSS.
- We compare the efficacy of THFSS-based RDTFs and optimal features survived by filter-and hybrid-oriented feature selection methods.

The rest of the paper is arranged as what follows. The THFSS is elaborated in Section 3. Experimental results related to exerting the THFSS on T28VTD and RDTFs-based TSP are given in Section 4. Furthermore, to prove the efficacy of THFSS, comparing the performance between THFSS and other feature selection algorithms is placed at the end of Section 4. Finally, the conclusion is depicted in Section 5.

3. Twin Hybrid FSS (THFSS)

Before elaborating on the proposed FSS based on the mathematical treatment presented in this section, we consider assumptions as follows:

- The learning scenarios decorated by mathematical treatment are exerted on transient space based on the *N-1* contingency criteria (where *N* refers to the number of elements in the network, and one refers to a single element that can fail in the power system) without regard for the contaminated transient data (noisy and missing transient data) and different operating conditions of the power grid.
- The transient dataset is balanced (balanced class).

The graphical summary of THFSS is shown in Figure 1. According to this figure, the primary materials in forming THFSS are based on two approaches: 1) filter: conditional relevancy rate (CRR) (CRR is based on mutual information (MI), and entropy calculations [12]), and 2) wrappers: incremental wrapper subset selection (IWSS) [17] and IWSS with replacement (IWSSr) [18]. The two steps of THFSS (Hyb^1) contain the filter-wrapper scheme. The first and second folds of THFSS are equipped with CRR-IWSS and CRR-IWSSr, respectively. For example, in the case of CRR-IWSS (hybrid of fold^1: Hyb^1), the CRR-based filter (green-face bullet) and IWSS-based wrapper (red-face bullet) are placed in the first to second steps of Hyb^1, respectively (see Figure 1). In filter, CRR is mounted on mutual information (MI) and entropy, and in wrappers, IWSS and IWSSr are triggered by kernelized support vector machine (SVM) [19] and twin SVM (TWSVM) [20] Kernelized hyperplane-based predictive models (KHPMs) situated in the wrapper of $\text{Hyb}^{1:2}$ for precise mining on transient nonlinear space cause arising IWSS- and IWSSr-based wrappers with four versions. Using radial basis function (RBF) [19] and dynamic time warping (DTW) [21] kernels in SVM and RBF and polynomial (POL) [22] kernels in TWSVM lead to forming $Wrapper_{RBF}^{SVM}$, $Wrapper_{DTW}^{SVM}$, $Wrapper_{RBF}^{TWSVM}$, and $Wrapper_{POL}^{TWSVM}$, which are connected successively (Wrapper: IWSS/ IWSSr). These connected KHPMs-based wrappers placed in each fold (second step of $\text{Hyb}^{1:2}$) are permuted in a 24-way manner (*factorial* of *4*) (see Figure 1; IWSS-based left permutations (LP^{IWSS}) in fold^1 and IWSSr-based right permutations (RP^{IWSSr}) in fold^2).

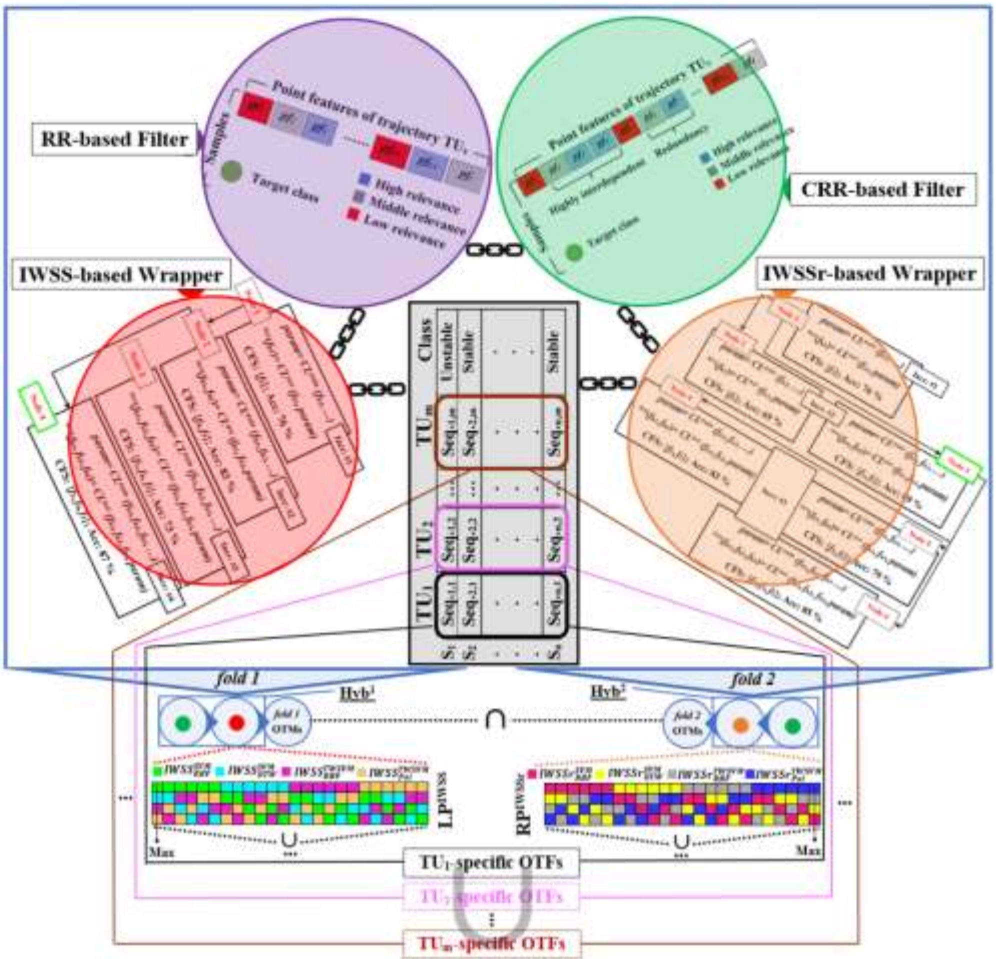

Figure 1. The Overall process of THFSS.

Finding RDTFs from transient m-time series data is realized by conducting THFSS on each transient univariate (TU) (see Figure 1; the transient data set was depicted as black-face TU_1, pink-face TU_2, ..., and brown-face TU_m, respectively). First, each TU_x is entered into the first step of $Hyb^{1:2}$ supported by the CRR-based filter, and then ${}^{1}Hyb^{1,2}_{CRR}TMs_x^{top\,k}$ is obtained. The CRR is formulated as (1) based on MI (2) and entropy (H) (3) formulations, which determine the information shared between $TU_x^{TM_i}$ and target class (C) in the presence $TU_x^{TM_j}$. This relationship can range from fully redundant to fully interdependent ($-1 \le CRR\,(TU_x^{TMs_i}, TU_x^{TMs_j}) \le 1$). For more information about CRR calculation, refer to Table 1. Next, TU_x-specific ${}^{1}Hyb^{1,2}_{CRR}TMs_x^{top\,k}$ is fed to KHPMs-based wrappers (${}^{2}Hyb^{1}$: IWSS-based wrapper and ${}^{2}Hyb^{2}$: IWSSr-based wrapper).

$$CRR(X,Y) = \frac{2(MI(TU_x^{(TMs_i)};C\,|\,TU_x^{(TMs_j)}) - MI(TU_x^{(TMs_i)};C))}{(H(TU_x^{(TMs_i)}) + H(C))} \quad (1)$$

where X and Y are $TU_x^{TMs_i}$ and $TU_x^{TMs_j}$, respectively.

$$MI(TU_x^{(TMs_i)};C) = H(TU_x^{(TMs_i)}) - H(TU_x^{(TMs_i)}\,|\,C) \quad (2)$$

$$H(Z) = -\sum_{z \in Z} p(z)\log p(z) \quad (3)$$

Based on exerting KHPMs-based IWSS and KHPMs-based IWSSr, $IWSS/IWSSr^{KHPMs}$ related to each permutation column in LP^{IWSS} or RP^{IWSSr} is recorded as the relevant transient features (${}^{2}Hyb^{1,2}_{\;x}RTFs_{per:p}^{IWSS/IWSSr}$).

In LP^{IWSS} or RP^{IWSSr}, the result of each $IWSS/IWSSr^{KHPMs}$ enters the $IWSS/IWSSr^{KHPMs}$ as input, and then a max function is applied to the quad-result of the permutation column. Next, the union function operates on ${}^{2}Hyb^{1,2}_{\;x}RTMs_{per:1}^{IWSS/IWSSr}$ to ${}^{2}Hyb^{1,2}_{\;x}RTMs_{per:24}^{IWSS/IWSSr}$, and union RTFs is recorded in ${}^{2}Hyb^{1:2}_{\;x}URTMs^{IWSS/IWSSr}$. After ending ${}^{2}Hyb^{1:2}$, the intersection operator is applied on $OTMs_x^{fold\;1:2}$ and the result is recorded as $OTMs_x$. Finally, applying union operation on $OTMs_1$ to $OTMs_m$ leads to surviving RDTFs.

Besides elaborating on the proposed FSS according to Figure 1 (see previous paragraph), the pseudocode of THFSS is given in Table 2. Also the complexity analysis of THFSS is presented for the readers. The complexity magnitude of THFSS is tightly correlated to the embedded wrapper mechanisms. The IWSS and IWSSr have O(n) and $O(n^2)$ complexity, respectively [18]. The complexity of KHPMs plugged into wrappers, namely SVM and TWSVM, is $O(n^3)$ and $O(2\times(n/2)^3)$, respectively [23]. Hence, $IWSS^{KHPMs}$ and $IWSSr^{KHPMs}$ have $O(\max\{(n\times n^3), (n\times 2\times(n/2)^3)\})$ and $O(\max\{(n^2\times n^3), (n^2\times 2\times(n/2)^3)\})$ complexity, respectively. The complexity ratio of SVM and TWSVM ($n^3 / (2\times(n/2)^3) = 4$) shows the fact that the complexity of $IWSS^{KHPMs}$ and $IWSSr^{KHPMs}$ is $O(n\times n^3)$ and $O(n^2\times n^3)$, respectively. Compacting the RHDTS and feeding the IWSSr tree with COTFS provide the necessary conditions for equality of complexity IWSSr by IWSS [18].

Table 1. The pseudo-code of the CRR.

Input: x^{th} univariate of transient trajectory data; {x=1, 2, …, m}.
Output: top k transient moments of TU_x based on CRR value.
(1) ${}^{TMs_{1:s}}RR_x$ = **RR** (TU_x^{TMs}); // For more information about RR function, refer to [11].
(2) ${}_{CRR}TMs_x^{top\,k} = \varnothing$;
(3) W_{TMs}^{0} = initial weight of TMs ∈ TU_x is set to one;
(4) TM^1 =find TM with highest RR in ${}^{TMs_{1:s}}RR_x$;
(5) ${}_{CRR}TMs_x^{top\,k} = {}_{CRR}TMs_x^{top\,k} \cup TM^1$;
(6) $TU_x^{TMs} = TU_x^{TMs} - TM^1$;
(7) **for** e=2 to k
(8) $W_{TMs}^{e-1} = W_{TMs}^{e-2} .* CRR\left(TU_x^{TMs_i}, TM^{e-1}\right)$; // CRR: See (1);
(9) ${}^{(TMs_{1:s}-TM^1)}RR_x^{Updated} = W_{TMs}^{e-1} .* \left({}^{(TMs_{1:s}-TM^1)}RR_x\right)$;
(10) TM^e =find TM with highest RR from ${}^{(TMs_{1:s}-TM^1)}RR_x^{Updated}$;
(11) ${}_{CRR}TMs_x^{top\,k} = {}_{CRR}TMs_x^{top\,k} \cup TM^e$;
(12) $TU_x^{TMs} = TU_x^{TMs} - TMs^e$;
(13) **end**
(14) **return** ${}_{CRR}TMs_x^{top\,k}$;

In such a situation, the complexity of $IWSS^{KHPMs}$ and $IWSSr^{KHPMs}$ is the same and will equal O(n^4). Consequently, the complexity of the proposed FSS is $O(c\times n^4)$.

Table 2. The pseudocode of the THFSS.

Main body of THFSS
Input: transient *m*-time series data (TD) ($\left[TD\right]_{n\times TU_{1:m}\ with\ L}$)
Output: Optimal transient moments (OTMs) of *m*-trajectory transient data.
(1) **for** x=1 to m
(2) $\ {}^{1}{}_{CRR}^{Hyb^{1:2}}TMs_x^{top\,k}$ = **CRR** (TU_x^{TMs}); Calculating the CRR value (See Table 1) of $TMs_i \in TU_x$ based on SU.
(3) **for** f=1:2 // filter-wrapper in a twofold way called Hyb$^{1:2}$ (fold$^{1:2}$).
(4) **if** f==1
(5) TMs=[ ];
(6) ${}^{2}{}_{Array}^{Hyb^{f}}$ = [**IWSS** (TMs, RBF, SVM), **IWSS** (TMs, DTW, SVM), **IWSS** (TMs, RBF, TWSVM), **IWSS** (TMs, POL, TWSVM)];
(7) **elseif** f==2
(8) TMs=[ ];
(9) ${}^{2}{}_{Array}^{Hyb^{f}}$ = [**IWSSr** (TMs, RBF, SVM), **IWSSr** (TMs, DTW, SVM), **IWSSr** (TMs, RBF, TWSVM), **IWSSr** (TMs, POL, TWSVM)];
(10) **end**
(11) **if** f==1 || f==2
(12) 4x24LWrapper$_x$ = **Perms** (${}^{2\,Hyb^{f}}Array$); //Perms: 'perms' command in MATALB environment.
(13) ${}^{2}{}_{Wrapper}^{Hyb^{f}}URTMs_x$ = **UnionLinkedWrappers** (4x24LWrapper$_x$, ${}^{1}{}_{CRR}^{Hyb^{1:2}}TFs_x^{top\,k}$);
(14) **end**
(15) **if** f==1
(16) $Rec_{OTMs}^{Hyb^{1:2}}$ =struct('*fold*', num2str(f), 'OTMs', ${}^{2}{}_{Wrapper}^{Hyb^{f}}URTMs_x$);
(17) **else**
(18) $Rec_{OTMs}^{Hyb^{1:2}}$ (end+1)=struct('*fold*', num2str(f), 'OTMs', ${}^{2}{}_{Wrapper}^{Hyb^{f}}URTMs_x$);
(19) **end**
(20) **end**
(21) $OTMs_x = \cap\ [\ Rec_{OTMs}^{Hyb^{1:2}}(1).OTMs : Rec_{OTMs}^{Hyb^{1:2}}(2).OTMs\ ]$.
(22) **if** x==1
(23) $OTMs^{Total}$ =struct('*Trajectory*', num2str(x), 'OTMs', $OTMs_x$);
(24) **else**
(25) $OTMs^{Total}$ (end+1)=struct('*Trajectory*', num2str(x), 'OTMs', $OTMs_x$);
(26) **end**
(27) **end**
(28) $RDTFs = \cup\ [\ OTMs^{Total}(1).OTMs : OTMs^{Total}(m).OTMs\ ]$.
Function: UnionLinkedWrappers (A, B)
(1) **for** p=1 to 24 // The number of permutations of the kernelized HPMs plugged into wrappers (4 factorial (4!)=24).
(2) $PerBox^{p}$ =A (:, p); $PerBox^{p}$ (1). arg^1=B;
(3) $RTMs_1^{p}$ =run ($PerBox^{p}$ (1)); B=Sort ($RTMs_1^{p}$);
(4) $PerBox^{p}$ (2). arg^1=B; $RTMs_2^{p}$ =run ($PerBox^{p}$ (2));
(5) B=Sort ($RTMs_2^{p}$); $PerBox^{p}$ (3). arg^1=B;
(6) $RTMs_3^{p}$ =run ($PerBox^{p}$ (3)); B=Sort ($RTMs_3^{p}$);
(7) $PerBox^{p}$ (4). arg^1=B; $RTMs_4^{p}$ =run ($PerBox^{p}$ (4));
(8) **if** p=1
(9) RTMs$^{1:24}$ =struct('*Permutation*', num2str(p), 'RTMs', max($RTMs_1^{p} : RTMs_4^{p}$));
(10) **else**
(11) RTMs$^{1:24}$(end+1) =struct('*Permutation*', num2str(p), 'RTMs', max($RTMs_1^{p} : RTMs_4^{p}$));
(12) **end**
(13) **end**
(14) $URTFs = \cup\ [\ RTMs^{1:24}(1).RTMs : RTMs^{1:24}(24).RTMs\ ]$.
(15) **return** $URTMs$;

* The pseudocode of the THFSS is rewritten based on MATLAB commands in the MATLAB environment.

4. Experimental Design

4.1. Selecting RDTFs from T28VTD

The transient data set (800 (No. transient cases) × 28 ($TU_{1:28}$) ×6 (No. observed cycles)) related to contingency simulation on the New England test system-New York power system (NETS-NYPS) is proposed by Canizares *et al.* [24] and generated by Python-SIEMENS PSS/E application program interface (API)-based code [25]. The list of transient 28-trajectory features is shown in Table 3. Also for more information about Python scripting for dynamic simulation based on PSS/E API, refer to [26]. After creating the transient dataset, finding RDTFs from $TU_{1:28}$ based on THFSS is on the agenda in this section according to Figure 1 and Table 2. After conducting the first step of $Hyb^{1:2}$ (${}^{1}Hyb^{1:2}$: CRR-based filter) on TU_x, which is called ${}^{1}Hyb^{1:2}_{CRR}TMs_x^{top\,k}$, the second step of $Hyb^{1:2}$ (KHPMs-based wrapper models) fed by ${}^{1}Hyb^{1:2}_{RR}TMs_x^{top\,k}$ is applied to each TU_x. The obtained results via ${}^{2}Hyb^{1:2}$ per TU_x is shown in Table 4. By completing the two steps of folds per TU_x, which is recorded in $Rec^{Hyb^{1:2}}_{OTMs_x}$ struct, the intersection function is operated on TU_x-specific twofold results (See Table 2, Line 21: ∩ [$Rec^{Hyb^{1:2}}_{OTMs}(1).OTMs : Rec^{Hyb^{1:2}}_{OTMs}(2).OTMs$]). The obtained results are shown in Table 5. The $OTMs^{Total}$ struct contains 28 OTMs-objects related to $TU_{1:28}$. Finally, applying the union operator on 28 records of $OTMs^{Total}$ struct (See Table 2, Line 28: ∪ [$OTMs^{Total}(1).OTMs : OTMs^{Total}(m).OTMs$]) cause to surviving RDTFs set. The members of the RDTFs set refer to the last row of Table 5.

Table 3. 28-trajectory transient features ($TU_{1:28}$)

Math formula
$TU_1^{t_m} = Max([\frac{PELEC_i}{P\max_i}]^{i=1:N_{genbus}})$
$TU_2^{t_m} = Var([\frac{PELEC_i}{P\max_i}]^{i=1:N_{genbus}})$
$TU_3^{t_m} = Max([\frac{QELEC_i}{Q\max_i}]^{i=1:N_{genbus}})$
$TU_4^{t_m} = Min([\frac{QELEC_i}{Q\max_i}]^{i=1:N_{genbus}})$
$TU_5^{t_m} = Var([\frac{QELEC_i}{Q\max_i}]^{i=1:N_{genbus}})$
$TU_6^{t_m} = Max([VOLT_i]^{i=1:N_{bus}})$
$TU_7^{t_m} = Var([VOLT_i]^{i=1:N_{bus}})$
$TU_8^{t_m} = Max([VANGLE_i]^{i=1:N_{bus}});\ \ slack\ bus = 0$

Table 3 (*Continued*.). 28-trajectory transient features ($TU_{1:28}$)

Math formula
$TU_9^{t_m} = Min([VANGLE_i]^{i=1:N_{bus}});\ \ slack\ bus = 0$
$TU_{10}^{t_m} = Var([VANGLE_i]^{i=1:N_{bus}}); slack\ bus = 0$
$TU_{11}^{t_m} = Max(abs([VANGLE_i - VANGLE_j]^{i,j=1:N_{bus}}))$
$TU_{12}^{t_m} = Mean(abs([VANGLE_i - VANGLE_j]^{i,j=1:N_{bus}}))$
$TU_{13}^{t_m} = Var(abs([VANGLE_i - VANGLE_j]^{i,j=1:N_{bus}}))$
$TU_{14}^{t_m} = \frac{\sum_{i=1}^{N_{busgen}} QLOAD_i}{\sum_{i=1}^{N_{busgen}} QELEC_i}$
$TU_{15:28}^{t_m} = Gradient\ of\ TU_1\ to\ TU_{14}$

Symbol: t_m= moments in simulation time [1: s], $N_{bus\ gen}$ = number of bus generator in test case, PELEC= machine electrical power (pu), P_{max}= maximum amount of machine electrical power, QELEC=machine reactive power, Qmax= maximum amount of machine reactive power, Qload= reactive power consumption, Volt= bus pu voltages, N_{bus}= number of buses in test case, VANGLE= voltage phase angle, Var= variance, Max= maximum, Min= minimum, Mean= average.

Table 4. The obtained results induced by exerting ${}^{2}Hyb^{1:2}$ (${}^{2}Hyb^{1}$: IWSS-based wrapper and ${}^{2}Hyb^{2}$: IWSSr-based wrapper) per TU_x.

Input	${}^{2}{}^{Hyb^1}_{IWSS}URTMs_x$	Input	${}^{2}{}^{Hyb^2}_{IWSSr}URTMs_x$
${}^{1}Hyb^{1}_{TU1}$	$\{{}^{1}TM_1, {}^{1}TM_{5:6}\}$	${}^{1}Hyb^{2}_{TU1}$	$\{{}^{1}TM_1, {}^{1}TM_{5:6}\}$
${}^{1}Hyb^{1}_{TU2}$	$\{{}^{2}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU2}$	$\{{}^{2}TM_{1:4}\}$
${}^{1}Hyb^{1}_{TU3}$	$\{{}^{3}TM_1, {}^{3}TM_5\}$	${}^{1}Hyb^{2}_{TU3}$	$\{{}^{3}TM_1, {}^{3}TM_{4:5}\}$
${}^{1}Hyb^{1}_{TU4}$	$\{{}^{4}TM_1, {}^{4}TM_{4:5}\}$	${}^{1}Hyb^{2}_{TU4}$	$\{{}^{4}TM_1, {}^{4}TM_{3:5}\}$
${}^{1}Hyb^{1}_{TU5}$	$\{{}^{5}TM_1, {}^{5}TM_{3,4}\}$	${}^{1}Hyb^{2}_{TU5}$	$\{{}^{5}TM_{1:2}, {}^{5}TM_4\}$
${}^{1}Hyb^{1}_{TU6}$	$\{{}^{6}TM_3\}$	${}^{1}Hyb^{2}_{TU6}$	$\{{}^{6}TM_{3:4}\}$
${}^{1}Hyb^{1}_{TU7}$	$\{{}^{7}TM_1, {}^{7}TM_3\}$	${}^{1}Hyb^{2}_{TU7}$	$\{{}^{7}TM_{1:2}\}$
${}^{1}Hyb^{1}_{TU8}$	$\{{}^{8}TM_{2:3}, {}^{8}TM_5\}$	${}^{1}Hyb^{2}_{U8}$	$\{{}^{8}TM_{2:3}, {}^{8}TM_5\}$
${}^{1}Hyb^{1}_{TU9}$	$\{{}^{9}TM_{1:3}\}$	${}^{1}Hyb^{2}_{TU9}$	$\{{}^{9}TM_{1:2}, {}^{9}TM_5\}$
${}^{1}Hyb^{1}_{TU10}$	$\{{}^{10}TM_{1:3}\}$	${}^{1}Hyb^{2}_{TU10}$	$\{{}^{10}TM_{1:3}\}$
${}^{1}Hyb^{1}_{TU11}$	$\{{}^{11}TM_1, {}^{11}TM_{3:4}\}$	${}^{1}Hyb^{2}_{U11}$	$\{{}^{11}TM_{1:4}\}$
${}^{1}Hyb^{1}_{TU12}$	$\{{}^{12}TM_{1:2}, {}^{12}TM_4\}$	${}^{1}Hyb^{2}_{TU12}$	$\{{}^{12}TM_{1:3}\}$
${}^{1}Hyb^{1}_{TU13}$	$\{{}^{13}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU13}$	$\{{}^{13}TM_{1:4}\}$
${}^{1}Hyb^{1}_{TU14}$	$\{{}^{14}TM_{1:3}\}$	${}^{1}Hyb^{2}_{TU14}$	$\{{}^{14}TM_{2:4}\}$
${}^{1}Hyb^{1}_{TU15}$	$\{{}^{15}TM_1\}$	${}^{1}Hyb^{2}_{TU15}$	$\{{}^{15}TM_{1:2}\}$
${}^{1}Hyb^{1}_{TU16}$	$\{{}^{16}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU16}$	$\{{}^{16}TM_{1:4}\}$
${}^{1}Hyb^{1}_{TU17}$	$\{{}^{17}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU17}$	$\{{}^{17}TM_{2:4}\}$
${}^{1}Hyb^{1}_{TU18}$	$\{{}^{18}TM_1, {}^{18}TM_{5:6}\}$	${}^{1}Hyb^{2}_{TU18}$	$\{{}^{18}TM_1, {}^{18}TM_{5:6}\}$
${}^{1}Hyb^{1}_{TU19}$	$\{{}^{19}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU19}$	$\{{}^{19}TM_{3:4}\}$
${}^{1}Hyb^{1}_{TU20}$	$\{{}^{20}TM_3, {}^{20}TM_5\}$	${}^{1}Hyb^{2}_{TU20}$	$\{{}^{20}TM_3, {}^{20}TM_5\}$
${}^{1}Hyb^{1}_{TU21}$	$\{{}^{21}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU21}$	$\{{}^{21}TM_1, {}^{21}TM_{3:4}\}$
${}^{1}Hyb^{1}_{TU22}$	$\{{}^{22}TM_{4:5}\}$	${}^{1}Hyb^{2}_{TU22}$	$\{{}^{22}TM_1, {}^{22}TM_{4:5}\}$
${}^{1}Hyb^{1}_{TU23}$	$\{{}^{23}TM_6\}$	${}^{1}Hyb^{2}_{TU23}$	$\{{}^{23}TM_6\}$
${}^{1}Hyb^{1}_{TU24}$	$\{{}^{24}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU24}$	$\{{}^{24}TM_{2:4}\}$
${}^{1}Hyb^{1}_{TU25}$	$\{{}^{25}TM_{2:3}, {}^{25}TM_{5:6}\}$	${}^{1}Hyb^{2}_{TU25}$	$\{{}^{25}TM_{2:3}, {}^{25}TM_{5:6}\}$
${}^{1}Hyb^{1}_{TU26}$	$\{{}^{26}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU26}$	$\{{}^{26}TM_{1:4}\}$
${}^{1}Hyb^{1}_{TU27}$	$\{{}^{27}TM_{1:2}, {}^{27}TM_4\}$	${}^{1}Hyb^{2}_{TU27}$	$\{{}^{27}TM_{1:4}\}$
${}^{1}Hyb^{1}_{TU28}$	$\{{}^{28}TM_{1:4}\}$	${}^{1}Hyb^{2}_{TU28}$	$\{{}^{28}TM_{3:4}\}$

x in ${}^{1}Hyb^{1,2}_{CRR}TMs_x^{top\,k}$: x^{th} TU; ${}^{x}TM_j$: j^{th} moments of TU_x, top 4: top 4 moments based on CRR.

Table 5. The obtained $OTMs_x$ set and RDTFs set.

Input	$OTMs_x$	Input	$OTMs_x$
TU_1: ∩ [.]	$\{^1TM_1, {}^1TM_{5:6}\}$	TU_{15}: ∩ [.]	$\{^{15}TM_1\}$
TU_2: ∩ [.]	$\{^2TM_{1:4}\}$	TU_{16}: ∩ [.]	$\{^{16}TM_{1:4}\}$
TU_3: ∩ [.]	$\{^3TM_1, {}^3TM_5\}$	TU_{17}: ∩ [.]	$\{^{17}TM_{2:4}\}$
TU_4: ∩ [.]	$\{^4TM_1, {}^4TM_4, {}^4TM_5\}$	TU_{18}: ∩ [.]	$\{^{18}TM_1, {}^{18}TM_{5:6}\}$
TU_5: ∩ [.]	$\{^5TM_1, {}^5TM_4\}$	TU_{19}: ∩ [.]	$\{^{19}TM_{3:4}\}$
TU_6: ∩ [.]	$\{^6TM_3\}$	TU_{20}: ∩ [.]	$\{^{20}TM_3, {}^{20}TM_5\}$
TU_7: ∩ [.]	$\{^7TM_1\}$	TU_{21}: ∩ [.]	$\{^{21}TM_1, {}^{21}TM_3, {}^{21}TM_4\}$
TU_8: ∩ [.]	$\{^8TM_2, {}^8TM_3, {}^8TM_5\}$	TU_{22}: ∩ [.]	$\{^{22}TM_4, {}^{22}TM_5\}$
TU_9: ∩ [.]	$\{^9TM_1, {}^9TM_2\}$	TU_{23}: ∩ [.]	$\{^{23}TM_6\}$
TU_{10}: ∩ [.]	$\{^{10}TM_{1:3}\}$	TU_{24}: ∩ [.]	$\{^{24}TM_{2:4}\}$
TU_{11}: ∩ [.]	$\{^{11}TM_1, {}^{11}TM_3, {}^{11}TM_4\}$	TU_{25}: ∩ [.]	$\{^{25}TM_2, {}^{25}TM_3, {}^{25}TM_5, {}^{25}TM_6\}$
TU_{12}: ∩ [.]	$\{^{12}TM_{1:2}\}$	TU_{26}: ∩ [.]	$\{^{26}TM_{1:4}\}$
TU_{13}: ∩ [.]	$\{^{13}TM_{1:4}\}$	TU_{27}: ∩ [.]	$\{^{27}TM_{1:2}, {}^{27}TM_4\}$
TU_{14}: ∩ [.]	$\{^{14}TM_2, {}^{14}TM_3\}$	TU_{28}: ∩ [.]	$\{^{28}TM_3, {}^{28}TM_4\}$
RDTFs: Union of $OTMs_1$ to $OTMs_{28}$ (∪ [$OTMs^{Total}(1).OTMs$: $OTMs^{Total}(28).OTMs$])			
$\{^1TM_{1,5:6}, {}^2TM_{1:4}, {}^3TM_{1,5}, {}^4TM_{1,4:5}, {}^5TM_{1,4}, {}^6TM_3, {}^7TM_1, {}^8TM_{2:3,5}, {}^9TM_{1:2}, {}^{10}TM_{1:3}, {}^{11}TM_{1,3:4}, {}^{12}TM_{1:2}, {}^{13}TM_{1:4}, {}^{14}TM_{2:3}, {}^{15}TM_1, {}^{16}TM_{1:4}, {}^{17}TM_{2:4}, {}^{18}TM_{1,5:6}, {}^{19}TM_{3:4}, {}^{20}TM_{3,5}, {}^{21}TM_{1,3:4}, {}^{22}M_{4:5}, {}^{23}TM_6, {}^{24}TM_{2:4}, {}^{25}TM_{2:3,5:6}, {}^{26}TM_{1:4}, {}^{27}TM_{1:2,4}, {}^{28}TM_{3:4}\}$			

[.]: $Rec_{OTMs}^{Hyb^{1:2}}(1).OTMs$: $Rec_{OTMs}^{Hyb^{1:2}}(2).OTMs$

4.2. TSP based on RDTFs set

Evaluating the performance of RDTFs in TSP is based on 10-fold cross-validation is considered in this section. Based on performance metrics (accuracy (Acc), True Positive Rate (TPR), and True Negative rate (TNR)), obtained results related to applying SVM^{RBF} on RDTFs per fold are given in Table 6. This table contains the Max (Acc), TPR, and TNR of folds, which are obtained via exerting fine-tuning-oriented train-test by SVM^{RBF}. Based on applying the mean function on folds' results, Acc 98.87 %, TPR 98.5 %, and TNR 99.25 % are obatined (see the last row of Table 6). For more clarity, the Acc variations in $fold^1$, $fold^3$, $fold^7$, and $fold^9$ are shown in Figure 2. In the 3-D charts of Figure 2, the X axis is related to the σ parameter variations ([-5:15]), the Y axis is related to C parameter variations ([0:15]), and the Z axis is related to the Acc fluctuations. Furthemore, processing time calculation based on transient observation cycles (TOCs) and prediction time is reported in this section. The members of RDTFs (see Table 5, last row) show the fact the maximum TOCs (MTOCs) are six cycles (e.g., $^1TM_6$, $^{18}TM_6$, and $^{23}TM_6$). Hence, 6 cycles are equal to 100.2 miliseconds (ms) (6×0.0167s (measurement rate)). In terms of prediction time, SVM^{RBF} labels to an unseen transient case in 2.387 ms. Consequently, the summation of 100.2 ms (MTOCs) and 2.453 ms (prediction time) gives the processing time (102.653 ms). Such a low processing time provides proper conditions for timely corrective control actions.

Table 6. Results of TSP based on RDTFs set.

Classifier	Test case	10-fold cross validation — Max(Acc.) per fold based on fine-tuning on SVM^{RBF} parameters — Accuracy [TPR / TNR]			
		fold 1	**fold 2**	**fold 3**	**fold 4**
		98.75 [97.5 / 100]	95 [95 / 95]	97.5 [95 / 100]	98.75 [100 / 97.5]
		fold 5	**fold 6**	**fold 7**	**fold 8**
SVM^{RBF}	NETS-NYPS	100 [100 / 100]	100 [100 / 100]	100 [100 / 100]	100 [100 / 100]
		fold 9		**fold 10**	
		98.75 [97.5 / 100]		100 [100 / 100]	
		Mean (measure) of folds: Accuracy [TPR / TNR]			
		98.87 [98.5 / 99.25]			

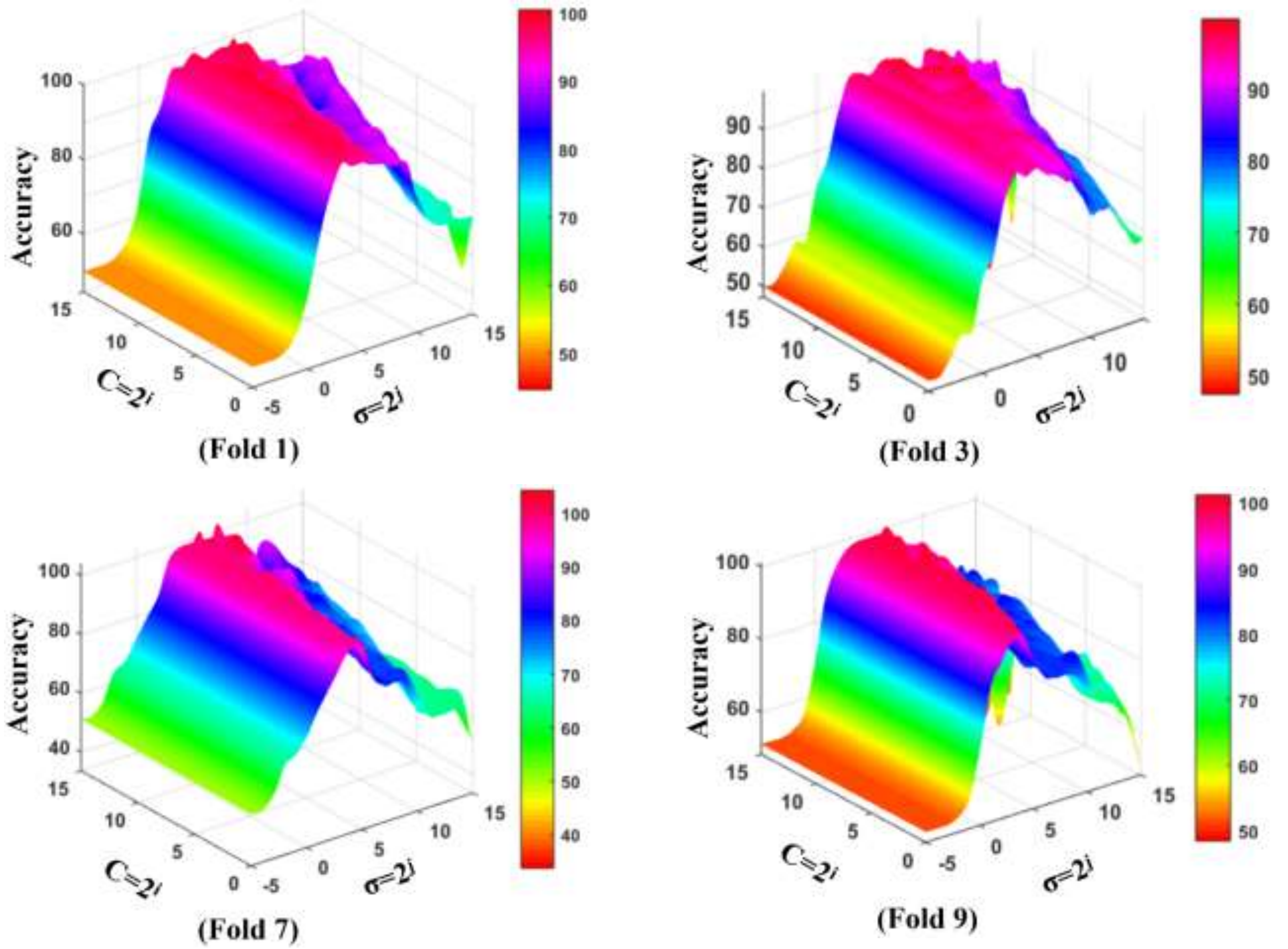

Figure 2. Acc variations in fold [1, 3, 7, 9] for TSP based on RDTFs set (In 3-D charts; X axis: *σ* parameter ([-5:15]), Y axis: *C* parameter ([0:15]), and Z axis: Acc fluctuations).

4.3 Comparison of experimental methods

In this section, comparing the proposed FSS with three filter-faced FSSs (3FFSSs) and two hybrid-faced (2HFSSs) is on the agenda. The mRMR [5], ReliefF [8], and FCBF [9] as 3FFSSs and BMHFSS [11] and PITHS [16] as 2HFSSs are compared with THFSS. After exerting 3FFSSs and 2HFSSs on T28VTD and entering survived optimal transient features (OTFs) into SVM^{RBF} in the same train-test procedure considered in our study, the obtained results show that THFSS^{RDTFs} have better performance in TSP than 3FFSSs^{OTFs} and 2HFSSs^{OTFs} (See Table 6 (last row) and Table 7). THFSS-based RDTFs containing 72 optimal cycles retrieved from T28VTD has better performance (Acc, TPR, and TNR) than mRMR^{OTFs} (9 OCs of T4VTD), FCBF^{OTFs}, ReliefF^{OTFs}, and BMHFSS^{OTMs} (9 OCs of T3VTD) [11], and PITHS^{OFs} (24 OCs of T28VTD [16]) (ignoring only 0.25% less than TPR than PITHS). From processing time aspect, SVM^{RBF}-THFSS^{RDTFs}-based processing time (102.653 ms) and the Table 8 results show that SVM^{RBF}-THFSS^{RDTFs} has a higher TPT (102.653 ms) than SVM^{RBF}-3FFSSs^{OTMs} (SVM^{RBF}-mRMR^{OTFs}: 68.793 ms, SVM^{RBF}-FCBF^{OTFs}: 68.930 ms, SVM^{RBF}-ReliefF^{OTFs}: 68.910 ms) and SVM^{RBF}-BMHFSS^{OTFs} (BMHFSS as 2HFSSs) with 52.948 ms. Also, SVM^{RBF}-THFSS^{RDTFs} have a lower processing time than SVM^{RBF}-PITHS^{OTFs} with 152.591 ms. The final report depicted in Table 8 (The seventh row of Table 8) indicates the amount of memory usage by the SVM^{RBF}-THFSS^{RDTFs} for TSP. For more information, refer to Table 8.

Table 7. Results of TSP Via coupling SVM^{RBF} and selected RDTFs by 3FFSSs and 2HFSSs.

Classifier-FSS	Test case	10-fold cross-validation: Max(Acc.) per fold based on fine-tuning on SVM^{RBF} parameters, Accuracy [TPR / TNR]			
SVM^{RBF}-mRMR	NETS-NYPS	**fold 1**	**fold 2**	**fold 3**	**fold 4**
		93.75 [97.5 / 90]	93.75 [97.5 / 90]	90 [95 / 85]	90 [82.5 / 97.5]
		fold 5	**fold 6**	**fold 7**	**fold 8**
		95 [97.5 / 92.5]	95 [95 / 95]	95 [100 / 90]	91.25 [87.5 / 95]
		fold 9		**fold 10**	
		88.75 [95 / 82.5]		93.75 [92.5 / 95]	
		Mean(measure) of folds: Accuracy [TPR / TNR]			
		92.62 [94 / 91.25]			
SVM^{RBF}-FCBF	NETS-NYPS	**fold 1**	**fold 2**	**fold 3**	**fold 4**
		98.75 [97.5 / 100]	95 [95 / 95]	96.25 [92.5 / 100]	97.5 [95 / 100]
		fold 5	**fold 6**	**fold 7**	**fold 8**
		96.25 [92.5 / 100]	97.5 [97.5 / 97.5]	97.5 [95 / 100]	97.5 [97.5 / 97.5]
		fold 9		**fold 10**	
		97.5 [95 / 100]		100 [100 / 100]	
		Mean(measure) of folds: Accuracy [TPR / TNR]			
		97.37 [95.75 / 99]			
SVM^{RBF}-ReliefF	NETS-NYPS	**fold 1**	**fold 2**	**fold 3**	**fold 4**
		98.75 [97.5 / 100]	95 [95 / 95]	96.25 [92.5 / 100]	97.5 [95 / 100]
		fold 5	**fold 6**	**fold 7**	**fold 8**
		96.25 [92.5 / 100]	97.5 [97.5 / 97.5]	97.5 [95 / 100]	97.5 [97.5 / 97.5]
		fold 9		**fold 10**	
		97.5 [95 / 100]		100 [100 / 100]	
		Mean(measure) of folds: Accuracy [TPR / TNR]			
		97.37 [95.75 / 99]			
SVM^{RBF}-BMHFSS	NETS-NYPS	**fold 1**	**fold 2**	**fold 3**	**fold 4**
		100 [100/ 100]	97.5 [100 / 95]	96.25 [92.5 / 100]	95 [92.5 / 97.5]
		fold 5	**fold 6**	**fold 7**	**fold 8**
		100 [100 / 100]	100 [100 / 100]	97.5 [97.5 / 97.5]	98.75 [97.5 / 100]
		fold 9		**fold 10**	
		97.5 [97.5 / 97.5]		100 [100 / 100]	
		Mean(measure) of folds: Accuracy [TPR / TNR]			
		98.25 [97.75 / 98.75]			
SVM^{RBF}-PITHS	NETS-NYPS	**fold 1**	**fold 2**	**fold 3**	**fold 4**
		98.75 [97.5 / 100]	96.25 [100 / 92.5]	97.5 [95 / 100]	98.75 [100 / 97.5]
		fold 5	**fold 6**	**fold 7**	**fold 8**
		100 [100 / 100]	100 [100 / 100]	98.75 [97.5/ 100]	100 [100 / 100]
		fold 9		**fold 10**	
		97.5 [97.5 / 97.5]		100 [100 / 100]	
		Mean(measure) of folds: Accuracy [TPR / TNR]			
		98.75 [98.75 / 98.75]			

Table 8. TSP based on 3FFSSs and 2HFSSs and amount of memory usage by SVM^{RBF}-$THFSS^{RDTFs}$ model.

SVM^{RBF}-3FFSSs / 2HFSSs	**MTOCs in cycle / second**	**Processing time (MTOCs + prediction time)**	
SVM^{RBF}-mRMR	4 / 0.0668	66.8 ms+1.993 ms= 68.793 ms	
SVM^{RBF}-FCBF	4 / 0.0668	66.8 ms+2.130 ms= 68.930 ms	
SVM^{RBF}-ReliefF	4 / 0.0668	66.8 ms+2.110 ms= 68.910 ms	
SVM^{RBF}-BMHFSS	3 / 0.0501	50.1 ms+2.848 ms= 52.948 ms	
SVM^{RBF}-PITHS	9 / 0.1503	150.3 ms+2.291 ms= 152.591 ms	
Memory status	**Max Possible Array Bytes**	**Mem available all arrays (MAAAs)**	**Mem used MATLAB**
Before starting model (^{BS}model)	4.0087e+09	4.0087e+09	1.3455e+09
After ending model (^{AE}model)	3.0849e+09	3.0849e+09	1.2750e+09
Memory used in Megabytes			
^{AE}Model.MAAAs – ^{BS}Model.MAAAs = - 881 Megabytes			

(-): indicates that the free memory is (about 251 megabytes) lower now than it was before started model (SVM^{RBF}-$THFSS^{OTFs}$).

5. Conclusions and Future Work

The RHDTS with IRNDTFs is a main obstacle in achieving fast-accurate TSA. Passing this obstacle is possible only through the feature selection process. Hence, in this paper, we offer the twin hybrid FSS (THFSS) to find RDTFs from T28VTD. The THFSS encompasses two folds in which the filter-wrapper scheme is executed. The filter steps are supported by CRR, and the wrapper step is triggered by IWSS and IWSSr mechanisms. The obtained results show that THFSS-based RDTFs have high performance (Acc 98.87 %, TPR 98.5 %, TNR 99.25 %, and transient processing time of 102.653 ms) on TSP. For evaluating the efficacy of the proposed FSS, THFSS is compared with other FSS. The results indicate that the THFSS-based RDTFs set is better than optimal features selected through other feature selection algorithms on TSP.

Selecting the most relevant features for high-performance TSP under the power grid's complicated conditions raised by *N-k* contingency, different load-generation levels, and contaminated transient samples (communication failure (unavailability) and lack of quality of power system dynamic responses (noisy data)) are main factors that are not considered in the design of our proposed learning framework. Hence, it is possible that our learning method cannot result in high-performance transient prediction as in normal transient data. Hence, this issue can be considered as the limitation of our proposed FSS. In the future FSS-based TSA, we decorate a convolutional neural network (CNN), in which extracted features by its layers feed the polyhedral feature selection algorithm. Such a scheme promises high-performance TSA under the power grid's complicated conditions raised by *N-k* contingency, different load-generation levels, and contaminated transient samples.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for profit sectors.

Notation

CRR (·)	conditional relevancy rate function.
RR (·)	relevancy rate function.
TU_x	x^{th} transient univariate (TU) of transient trajectory data.
TU_x^{TMs}	transient moments (TMs) in TU_x.
${}^{TMs_{1:s}}RR_x$	sorted TMs of TU_x via RR function.
${}_{CRR}TMs_x^{top\,k}$	recording top TMs based on CRR formula per TU_x.
W_{TMs}^{0}	initial weight of TMs ∈ TU_x is set to one.
TM^1	*TM* with highest RR in ${}^{TMs_{1:s}}RR_x$.
W_{TMs}^{e-1}	*e-1*th weight of TMs ∈ TU_x calculated based on CRR function.
$(TMs_{1:s} - TM^1)RR_x^{Updated}$	updating the weights of TMs ∈ TU_x except TM^1.
TM^e	*TM* with highest RR from $(TMs_{1:s} - TM^1)RR_x^{Updated}$
$[TD]_{n\times TU_{1:m}\ with\ L}$	transient data (TD) with *n* sample, *m* trajectory features, and labels (L)

${}^{1}{}_{CRR}^{Hyb^{1:2}}TMs_{x}^{top\,k}$	recording top TMs based on CRR formula per TU_x by exerting the first step of fold[1] and fold[2] of THFSS
${}^{2}{}_{Array}^{Hyb\,f}$	The array contains the IWSS/IWSSr-based learning model accompanied by kernelized SVM and TWSVM, which are situated in the second step of folds.
$4x24LWrapper_x$	data structure for recording the results of various permutations in execution ${}^{2}{}_{Array}^{Hyb\,f}$.
${}^{2}{}_{Wrapper}^{Hyb\,f}URTMs_x$	the union of obtained results recorded in $4x24LWrapper_x$ (Called union of relevant transient moments of TU_x ($URTMs_x$)).
$Rec_{OTMs}^{Hyb^{1:2}}$	a struct for recording results of ${}^{2}{}_{Wrapper}^{Hyb\,f}URTMs_x$ per fold.
$OTMs_x$	optimal transient moments of TUx derived by intersecting the obtained results of fold[1]-specific $Rec_{OTMs}^{Hyb^{1:2}}$ and fold[2]-specific $Rec_{OTMs}^{Hyb^{1:2}}$ per TU_x.
$OTMs^{Total}$	a struct for recording OTMs set per TU_x.
RDTFS	relevant-discriminative transient features derived by the union of $OTMs_{1\ to}$ $OTMs_{28}$.
$PerBox^{p}$	different permutations of IWSS/IWSSr-based learning scenarios accompanied with kernelized SVM and TWSVM decorated in 24 manners (*p*: 1 to 24) situated in the second step of fold[1] and fold[2].
$RTMs_{N^{th}}^{p}$	obtained results (relevant transient moments (RTMs)) derived by execution of N^{th} rounds of $PerBox^{p}$ (N^{th}) (rounds from 1 to 4).
$RTMs^{1:24}$	a struct contains RTMs derived by applying max function on $RTMs_1^p$ to $RTMs_2^p$ per *p*.
URTFs	the union relevant transient features (URTFs) stemmed from union of $RTMs^{1:24}(1)$ to $RTMs^{1:24}$ (24).

References

[1] Khosravi, M., Banejad, M. and Toosian Shandiz, H., "Robust state estimation in power systems using pre-filtering measurement data", Journal of AI and Data Mining, vol. 5, no. 1, pp. 111-125. 2017.

[2] J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques", 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.

[3] P. Kundur et al., "Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definition," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1387–1401, Aug. 2004.

[4] M. Pavella, M. Ernest, and D. Ruiz-Vega, "Transient Stability of Power Systems: A Unified Approach to Assessment and Control", 1st ed. New York, NY, USA: Springer, 2000.

[5] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An Introduction to Statistical Learning", 1st ed. New York, NY, USA: Springer, 2013.

[6] X. Li, Z. Zheng, L. Wu, R. Li, J. Huang, X. Hu, and P. Guo, "A stratified method for large-scale power system transient stability assessment based on maximum relevance minimum redundancy arithmetic," *IEEE Access*, vol. 7, pp. 61414–61432, May 2019.

[7] J. Liu, H. Sun, Y. Li, W. Fang, and S. Niu, "An improved power system transient stability prediction model based on mRMR feature selection and WTA ensemble learning," *Appl. Sci.*, vol. 10, no. 7, p. 2255, Mar. 2020.

[8] A. Stief, J. R. Ottewill, and J. Baranowski, "Relief F-based feature ranking and feature selection for monitoring induction motors," *in Proc. 23rd Int. Conf. Methods Models Autom. Robot. (MMAR)*, Miedzyzdroje, Poland, pp. 171–176, Aug. 2018.

[9] J. Yan, C. Li, and Y. Liu, "Deep learning based total transfer capability calculation model," *in Proc. Int. Conf. Power Syst. Technol. (POWERCON)*, Guangzhou, China, pp. 952–957, Nov. 2018.

[10] L. Ji, J. Wu, Y. Zhou, and L. Hao, "Using trajectory clusters to define the most relevant features for transient stability prediction based on machine learning method," *Energies*, vol. 9, no. 11, p. 898, Nov. 2016.

[11] Z. Chen, X. Han, C. Fan, T. Zheng, and S. Mei, "A two-stage feature selection method for power system transient stability status prediction," *Energies*, vol. 12, no. 4, p. 689, Feb. 2019.

[12] S. A. Bashiri Mosavi, "Extracting most discriminative features on transient multivariate time series by bi-mode hybrid feature selection scheme for transient stability prediction," *IEEE Access*, vol. 9, pp. 121087–121110, Aug. 2021.

[13] Y. Li and Z. Yang, "Application of EOS-ELM with binary Java-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23092–23101, Oct. 2017.

[14] X. Gu, Y. Li, and J. Jia, "Feature selection for transient stability assessment based on kernelized fuzzy rough sets and memetic algorithm," *Electrical Power and Energy Systems*, vol. 64, pp. 664–670, Jan. 2015.

[15] S. A. Bashiri Mosavi, "Applying cross-permutation-based quad-hybrid feature selection algorithm on transient univariates to select optimal features for transient analysis," *IEEE Access*, vol. 10, pp. 41131–41151, Apr. 2022.

[16] S. A. Bashiri Mosavi, "Finding Optimal Point Features in Transient Multivariate Excursions by Horizontally Integrated Trilateral Hybrid Feature

Selection Scheme for Transient Analysis," *IEEE Access*, vol. 9, pp. 163297-163315, Dec. 2021.

[17] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Incremental wrapper-based gene selection from microarray data for cancer classification," *Pattern Recognition*, vol. 39, no. 12, pp. 2383-2392, Dec. 2006.

[18] P. Bermejo, J. A. Gámez, J. M. Puerta, "Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection," *In: Proc. IEEE Symposium Series on Computational Intelligence and Data Mining (CIDM)*, Nashville, USA, pp. 367–374, May 2009.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, Apr. 1995.

[20] Javadeva, R. Khemchandani, and S. Chandra, "Twin Support Vector Machines for Pattern Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, May 2007.

[21] H. Shimodaira, K.-I. Noma, M. Nakai, and S. Sagayama, "Support vector machine with dynamic time-alignment kernel for speech recognition," *in Proc. Eur. Conf. Speech Commun. Technol. (EUROSPEECH)*, Aalborg, Denmark, pp. 1–4, 2001.

[22] C. M. Bishop, "Pattern Recognition and Machine Learning", 1st ed. NY: Springer, 2006.

[23] D. Tomar and S. Agarwal, "Twin support vector machine," *Egyptian Informat. J.*, vol. 16, no. 1, pp. 55–69, Mar. 2015.

[24] C.Canizares, T. Fernandes, E. Geraldi, Jr., L. Gérin-Lajoie, M. Gibbard, I. Hiskens, J. Kersulis, R. Kuiava, L. Lima, F. de Marco, N. Martins, B. Pal, A. Piardi, R. Ramos, J. dos Santos, D. Silva, A. Singh, B. Tamimi, and D. Vowles, "Benchmark systems for small-signal stability analysis and control," IEEE Power Energy Soc., Piscataway, NJ, USA, Tech. Rep. PESTR18, Aug. 2015.

[25] Siemens Industry, Schenectady, NY, USA. (2013). *Siemens Power Technologies International, PSS/E 33.4 Application Program Interface (API)*. [Online]. Available: https://www.siemens.com/power-technologies.

[26] S.A. Bashiri Mosavi, "Extracting discriminative features n reactive power variations by 1-persistance parallel fragmented hybrid feature selection scheme for transient stability prediction," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 500–513, Aug. 2021.

بکارگیری طرح انتخاب ویژگی دوگانه-ترکیبی بر روی داده‌های چند مسیره‌ی گذرا جهت پیش‌بینی پایداری گذرا

سید علیرضا بشیری موسوی[1*] و امید خلف بیگی[2]

[1]گروه مهندسی برق و کامپیوتر، مرکز آموزش عالی فنی و مهندسی بوئین زهرا، بوئین زهرا، قزوین، ایران.

[2]گروه مهندسی برق و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.

ارسال ۲۰۲۳/۰۹/۰۲؛ بازنگری ۲۰۲۳/۱۰/۱۲؛ پذیرش ۲۰۲۳/۱۰/۳۱

چکیده:

یک ارزیابی سریع و دقیق پایداری گذرا با ا ستفاده از الگوریتم‌های کارآمد یادگیری ما شین و مبتنی بر آمار در ف ضای سری زمانی غیرخطی گذرا به د ست می‌آید. فرآیند انتخاب ویژگی با تشـکیل فضـای ویژگی گذرا فشـرده از داده‌های گذرا با ابعاد بالا می‌تواند راه را برای ارزیابی پایدرای گذرا با کارایی بالا هموار کند. از این رو، طراحی یک طرح جامع انتخاب ویژگی که بتواند فضای داده ای گذرا را با ویژگی‌های گذرای متمایز مرتبط پر کند، یک نیاز فوری است. هدف این کار معرفی ساختار ترکیبی دوقلو برای انتخاب ویژگی های بهینه از داده‌های سری زمانی ۲۸ متغیره است. هر بخش از الگوریتم پیشنهادی شامل مکانیزم های فیلتر و پو ششی ا ست. نرخ وابستگی مشروط، بر ا ساس اطلاعات متقابل و محا سبات آنتروپی، به عنوان روش فیلتر در نظر گرفته می شود و انتخاب زیر مجموعه پوششی افزایشی و نسخه با جایگزین این روش که توسط ماشین بردار پشتیبان هسته دار و نسخه دو قلو آن تغذیه می‌شود، به عنوان فاز پوششی اســتفاده می‌شــود. پس از اعمال روش پیشــنهادی بر روی تک متغیره‌های گذرا، ویژگی‌های منتخب وارد روش آزمایش مبتنی بر اعتبارســنجی متقابل برای ارزیابی کارایی آنها در ارزیابی پایداری گذرا می شوند. نتایج بد ست آمده ن شان می دهد که ویژگی‌های منتخب مبتنی بر روش پی شنهادی دارای دقت پیش بینی ۹۸,۸۷ درصد و زمان پردازش ۱۰۲,۶۵۳ میلی ثانیه برای پیش بینی وضعیت پایداری گذرا هستند.

کلمات کلیدی: طرح انتخاب ویژگی ترکیبی، ویژگی‌های گذرای متمایز مرتبط، پیش‌بینی پایداری گذرا.