

RESEARCH ARTICLE

Surviving Most Relevant Features on Transient Trajectories Data by Dyadic 24-Way Hybrid Feature Selection Algorithm for Transient Stability Prediction

SEYED ALIREZA BASHIRI MOSAVI 

Department of Electrical and Computer Engineering, Buein Zahra Technical University, Buein Zahra, Qazvin 3451866391, Iran


e-mail: abashirimosavi@bzte.ac.ir

ABSTRACT Designing an effective feature selection scheme (FSS) is an inevitable solution for top-level balancing contrastive-correlated indices, namely transient processing time (TPT) and transient prediction accuracy (TPA) on transient stability assessment (TSA). Achieving low TPT and high TPA have a tight relationship in selecting the most relevant transient point features (MRTPFs) survived by applying comprehensive FSS on m -variate transient trajectory features (m VTTFs). Hence, we introduce dyadic 24-way hybrid FSS (D24WHFSS) to select MRTPFs from m VTTFs. The D24WHFSS comprises 24 permutations of the chained four-stage hybrid structure called 24-way hybrid FSS (24WHFSS). The 24WHFSS raised by bi-incremental wrapper mechanism (bi-IWM) contains incremental wrapper subset selection (IWSS) and IWSS with replacement (IWSSr). Each hybrid scenario is equipped with symmetric uncertainty (SU) (filter phase) and dual support vector-based classifiers (DSVCs) (wrapper phase). Embedded DSVCs into IWSS/ IWSSr include kernel support vector machine (k SVM) and k -twin SVM (k TWSVM). By plugging dual kernel function pairs (DKFPs) into DSVCs, 24-way $SU^{bi-IWM}DSVCs$ is exerted in the varied twofold repetition (dyadic 24WHFSS). In the first KFP (KFP¹), the radial basis function (RBF) is situated in the DSVCs of bi-IWM. In KFP², the dynamic time warping (DTW) and polynomial (Poly) kernels are used in 24-way $SU^{bi-IWM}DSVCs$ that the DTW and Poly kernels plugged into $SU^{IWSS^kSVM}/SU^{IWSSr^kSVM}$ and $SU^{IWSS^kTWSVM}/SU^{IWSSr^kTWSVM}$, respectively. Finally, the efficacy of D24WHFSS-based MRTPFs in TSA is evaluated via cross-validation. The results show that D24WHFSS has a TPA of 99.25 % and a TPT of 102.607 milliseconds for TSA.

INDEX TERMS Hybrid feature selection scheme, most relevant transient point features, transient stability assessment.

ACRONYMS

DM	Data Mining.	TPT	Transient Processing Time.
FSP	Fast-Sudden Phenomena.	TPA	Transient Prediction Accuracy.
HPA	High Prediction Accuracy.	IRTFs	Irrelevant-Redundant Transient Features.
LPT	Low Processing Time.	HDTs	High Dimensional Transient Space.
HDS	High Dimensional Space.	FSS	Feature Selection Scheme.
IRFs	Irrelevant and Redundant Features.	MRTPFs	Most Relevant Transient Point Features.
FS	Feature Selection.	MLCs	Machine Learning Classifiers.
TSA	Transient Stability Assessment.	CTS	Compact Transient Space.
PMUs	Phasor Measurement Units.	ITPs	Information Theory Principles.
		MI	Mutual Information.
		mRMR	Minimum-Redundancy and Maximum-Relevance.
		FCBF	Fast Correlation-Based Filter.

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Li .

TTC	Transfer Capability Calculation.
SVM	Support Vector Machine.
NMI	Normalized Mutual Information.
BPSO	Binary Particle Swarm Optimization.
FICA	Fuzzy Imperialist Competitive Algorithm.
IWSS	Incremental Wrapper Subset Selection.
MWFWs	Mono-Way Filter-Wrapper Structure.
IMRTPFs	Invisible MRTPFs.
mVTTFs	m-Variate Transient Trajectory Features.
SWSD	Single-Window Streaming Data.
TSP	Transient Stability Prediction.
D24WHFSS	Dyadic 24-Way Hybrid FSS.
bi-IWM	bi-Incremental Wrapper Mechanism.
SU	Symmetric Uncertainty.
TU	Transient Univariate.
PSS/E	Power System Simulator for Engineering.
MATLAB	Matrix Laboratory.
IWSSr	IWSS with Replacement.
DSVCs	Dual Support Vector-based Classifiers.
kSVM	kernel Support Vector Machine.
kTWSVM	kernel Twin Support Vector Machine.
DKFPs	Dual Kernel Function Pairs.
RBF	Radial Basis Function.
Poly	Polynomial.
DTW	Dynamic Time Warping.
TPFs	Transient Point Features.
L4SH	Linked 4-Stage Hybrid.
6L4SH	Six-type of L4SH.
OFs	Optimal Features.
ITIs	Information Theory-based Indices.
3FPs	Triple Fundamental Principles.
GEPSVM	Generalized Proximal Eigenvalue SVM.
KKT	Karush-Kuhn-Tucker.
2STDCM	Two-Step Transient Data Creation Mechanism.
VOLT	Bus Voltages.
VANGLE	Voltage Phase Angle.
PELEC	Machine Active Power.
QELEC	Machine Reactive Power.
QLOAD	Reactive Power Consumption.
PCS	Python-based Contingency Simulation.
API	Application Program Interface.
NETS-NYPS	New England Test System-New York Power System.
CONL	Convert Load.
Acc	Accuracy.
TPR	True Positive Rate (Sensitivity).
TNR	True Negative Rate (Specificity).
OWT	Observation Window Time.
3MWHFSSs	three Mono-Way Hybrid FSSs.
3MCWHFSSs	three Multi-Circular-Way Hybrid FSSs.
BMHFSS	Bi-Mode Hybrid FSS.
CPQHFSS	Cross-Permutation-based Quad-Hybrid FSS
PITHS	Partial-Injective Trilateral Hybrid Scheme.

NOTATION

f_i	i^{th} feature.
f_{hi}	Feature with i^{th} -highest SU.
$Acc(f_{hi})$	Learning model accuracy based on f_{hi} .
Incr: # i	i^{th} increments in IWSS/ IWSSr tree.
$CL^{train}(f_i)$	Training procedure of classification learner based on f_i .
param	Recording learning parameters.
$CL^{test}(f_i, param)$	Testing procedure of model based on param and f_i .
$OCTV_s^{BF_i}$	Output channels transient values related to i^{th} basic features; i : {VOLT/VANGLE/PELEC/QELEC/QLOAD}.
TU_x	x^{th} transient univariate in mVTTFs.
xMRTPFs	MRTPFs subset related to TU_x .
$UMRTPFs^{1:m}$	Union result of 1MRTPFs to mMRTPFs .
$TU_x^{TPFs_j}$	j^{th} TPFs of x^{th} TU.
$H(X)$	Entropy of X; X:{ $TU_x^{TPFs_j}$, target class}.
Max	Maximum function.
Min	Minimum function.
Var	Variance function.
$RankTPFs TU_x$	Ranked TPFs of TU_x based on SU values.
$Z^zRTPFs_x^i$	Contain the Zone ^z -specific RTPFs per iD24WHFSS .
$Z^{1:4}URTPFs_x^i$	Contain union result of Zone ¹ -Specific RTPFs to Zone ⁴ -specific RTPFs per iD24WHFSS .
$Z^{1:4}MRTPFs_x^{1:4}$	Contain union result of obtained 1D24WHFSS -specific CRTPFs and 2D24WHFSS -specific CRTPFs.
$DKFP^i$	i^{th} kernel function pair; i : {(RBF, RBF)/(DTW, Poly)}.
OFs_{total}^{stages}	Struct for recording the L4SH ^{1:6} -specific OFs related per zone of iD24WHFSS .

I. INTRODUCTION

Nowadays, the application of data mining (DM) technologies [1], [2] for promoting the prediction quality of fast-sudden phenomena (FSP) in core strategic industries (e.g., the energy industry, health industry, transportation industry, and so on) leads to realizing intelligent insights for system stakeholders [3], [4], [5]. Measuring FSP prediction quality depends on contrastive-correlated metrics, namely the accuracy of system status prediction and processing time for system status labeling. Achieving high prediction accuracy (HPA) and low processing time (LPT) simultaneously is defined ultimate goal in supervision-required industries. Such targeting in cognition of system phenomena markedly reduces system operator's directing challenges in conducting the timely-accurate intervention actions to return the system to a normal operation

state. However, a significant concept called high dimensional space (HDS) negatively overshadows the performance of DM-based tools to reach HPA and LPT-based effective decision-making on the system under study. The HDS arises from massive variables observed by software and hardware-oriented monitoring systems that irrelevant and redundant features (IRFs) populate a considerable portion of HDS. The presence of IRFs in DM-based extracting pattern procedures reduces prediction accuracy and increases the processing time related to unseen case labeling. Hence, the problem of dimensionality is a critical topic in pattern recognition [6], [7]. The right solution presented by DM experts to handle the HDS is termed the feature selection (FS) process [8], [9]. Applying FS-based techniques on HDS be caused discarding IRFs and surviving the most relevant ones. The selected subset encompasses features with minimum redundancy and maximum relevance criteria that promise low processing time and high accuracy prediction regarding FSP occurred in high-risk industries.

The energy sector as a sensitive industry includes the strategic product, namely electricity. Electric power has an incomparable role in economic prosperity and meeting human needs in modern society by guaranteeing the survival and continued growth of downstream to upstream industries. Hence, power system operational reliability assessment is a 24/7 supervisory support to ensure a stable power supply. One of the significant branches of dynamic stability assessment of the power grid is transient stability assessment (TSA) [10]. TSA aims to predict the transient stability status based on the data analytics dashboard raised by coupling DM technologies and transient data obtained by phasor measurement units (PMUs) [11] so that the system operator triggers a prompt-correct reaction against the unstable state. However, on the way to the synchronal achievement of low transient processing time (TPT) (low prediction time and small observation window) and high transient prediction accuracy (TPA) on TSA, the irrelevant-redundant transient features (IRTFs) in high dimensional transient space (HDTS) is the main obstacle. To solve this concern, applying the feature selection scheme (FSS) on HDTS to select the most relevant transient point features (MRTPFs) should be on the agenda. The IRTFs elimination and streaming of more information between MRTPFs and target class cause facilitate the well-training process condition of machine learning classifiers (MLCs) for high TPA. On the other hand, MRTPFs-based compact transient space (CTS) brings low TPT (low prediction time due to CTS-based faster training procedure; and selecting small observation window from MRTPFs subset), which causes to pass the time constraint in demanded corrective actions [12]. Considering the above points, designing the effective FSS for high-performance TSA has become a hot topic for DM specialists.

II. RELATED WORKS

Scrutinizing glance at the FS-based TSA studies manifests that applied FS mechanisms to find optimal transient fea-

tures are mounted on information theory principles (ITPs) (filter) and ITPs-MLCs approaches (filter-wrapper). In the case of filter-oriented works like [13] and [14], power and angle-based HDTS is targeted by mutual information (MI) metric to gain the best quantify of redundancy and relevance (minimum-redundancy and maximum-relevance (mRMR)). Another way of measuring the relevancy of observed features for monitoring the induction motor is the extended Relief called ReliefF elaborated in [15]. In [16], the fast correlation-based filter (FCBF) is situated as the primary stage in the transfer capability calculation (TTC) model, which by selecting optimal features helped the grid operators in addressing triple issues, namely static security, static voltage stability, and transient stability. In the case of the hybrid FS frameworks used in transient studies, [17] proposes the Relief and support vector machine (SVM)-based filter-wrapper combination to extract the optimal features on transient trajectories data set. In [18], hybrid FSS is appeared by integrating the normalized mutual information (NMI) (filter phase) and binary particle swarm optimization (BPSO) (wrapper phase) for high-performance transient stability status prediction. To surmount the HDTS, [19] presents the point and trajectory-feeding hybrid algorithm in the form of coupling fuzzy imperialist competitive algorithm (FICA) and incremental wrapper subset selection (IWSS) called FICA-IWSS that includes MI and conditional MI metrics in the filter phase and kernel SVM in the wrapper phase. In [20], a BinJava-based kernelized fuzzy rough sets (KFRS) approach is conducted on the entire feature space for selecting optimal feature subsets. In [21], coupling the kernelized fuzzy rough sets (KFRS) and the memetic algorithm is applied to transient data to survive the optimal transient features for TSA of power systems. In [22], cross-permutation-based quad-hybrid FSS (CPQHFSS) to select optimal features from TMEs. The CPQHFSS consists of four filter-wrapper blocks (FWBs) in the form of twin two-FWBs mounted on two-mechanism of the incremental wrapper. Reference [23] presents the partial-injective trilateral hybrid scheme (PITHS) based on horizontally integrated mode is applied on transient multivariate trajectory features (TMTFs) which consist of two nested trilateral phases namely nested trilateral filter phase (NTFP) and the nested trilateral wrapper phase (NTWP).

Focusing on the past FS-based TSA studies (e.g., [13], [14], [15], [16], [17], [18], [19], [22], and [23]) revealed the released strategies suffer from the mono-way filter-wrapper structure (MWFWS) that causes failure in the precise exploring MRTPFs from nonlinear HDTS. Passing the weak-learner MWFWS gates requires designing the well-structured FSS supported by a multi-level circular learning model (MLCLM). Performing MLCLM on foggy non-separable transient data brings the retrieving of invisible MRTPFs (IMRTPFs). On the other hand, in some of FS-based TSA like [20] and [21], applying MLCLM on m -variate transient trajectory features (m VTTFs) set to single-window streaming data (SWSD) mode stemmed from

sticking transient univariates together leads to pruning features defeated by selected features according to filter-wrapper metrics (a slight distinction between discarded features and optimal ones). In this regard, replacing the sectoral-oriented view with the SWSD mode is compelling in the feature selection process. Generally, overcoming mentioned obstacles has a direct impact on achieving timely-accurate transient stability prediction (TSP).

The trilateral contributions of this paper to handle FS-based TSP problems faced by transient analysts are categorized as follows:

- A novel feature selection algorithm named the dyadic 24-way hybrid FSS (D24WHFSS) is proposed to extract optimal features for high-performance TSA. The offered scheme including the linked four-stage hybrid model in a 24-way manner (called 24WHFSS) mounted on the bi-incremental wrapper mechanism (bi-IWM). The bi-IWM is decorated by symmetric uncertainty (SU)-based filter phase and hyperplane-based MLCs as wrapper phase. To reach CTS containing discriminative transient features, HDTs is fed to kernel-based in the varied twofold-repetition of 24WHFSS (dyadic 24WHFSS).
- Based on the SWSD in FS, the univariates of $mVTTFs$ are fed to the D24WHFSS separately. Besides extracting univariate-specific MRTPFs in such an approach, the risk of discarding optimal features in the streamed feature set induced by pasting together features of transient univariates (TUs) (TU_1 to TU_m) will be close to zero.
- The performance of D24WHFSS-based MRTPFs in TSP is compared with survived MRTPFs by various FS algorithms via cross-validation.

The rest of the article is structured as follows: The proposed D24WHFSS are elaborated in Section 3. Experimental results of exerting D24WHFSS on $mVTTFs$ and MRTPFs-based TSP are depicted in Section 4. Also, Section 4 ending is explained the performance comparison of the D24WHFSS-based MRTPFs with the optimal features survived by other FS techniques in TSA. Finally, the conclusion is remarked in Section 5.

III. DYADIC 24-WAY HYBRID FEATURE SELECTION SCHEME (D24WHFSS)

The overall workflow for high-performance TSA centered on D24WHFSS is depicted in Fig. 1. As the preliminary step, the contingency simulation to construct transient data set is performed by a triad of SIEMENS power system simulator for engineering (PSS/E) software, Python technology, and matrix laboratory (Matlab) tools. Next, dimensionality reduction of HDTs by introducing D24WHFSS is on the agenda. The proposed hybrid FSS is driven by 24 permutations of the chained four-stage hybrid models mounted on bi-IWM called 24WHFSS. The bi-IWM encompasses incremental wrapper subset selection (IWSS) and IWSS with replacement (IWSSr), which is supported by SU and

dual support vector-based classifiers (DSVCs) as the filter and wrapper phases of D24WHFSS, respectively ($filter$ bi-IWMS wrapper including SU IWSS DSVCs and SU IWSSr DSVCs). The DSVCs contain kernel support vector machine ($kSVM$) and k -twin SVM ($kTWSVM$). Based on setting dual kernel function pairs (DKFPs) into DSVCs, 24WHFSS is conducted in varied twofold repetition (dyadic 24WHFSS). Each KFP contains two functions: the first function is the bi-IWM kSVM -specific kernel and the second function is the bi-IWM kTWSVM -specific kernel. As the first KFP (KFP 1), the radial basis function (RBF) kernel plugged into DSVCs of bi-IWM that we have four-stage hybrid scenarios including SU IWSS RBFkSVM , SU IWSS RBFkTWSVM , SU IWSSr RBFkSVM , and SU IWSSr RBFkTWSVM . In KFP 2 , SU IWSS kSVM _ SU IWSSr kSVM -specified kernel is dynamic time warping (DTW) and SU IWSS kTWSVM _ SU IWSSr kTWSVM -specified kernel is polynomial (Poly) kernel. Finally, the efficacy rate of survived MRTPFs by D24WHFSS in TSP is measured via cross-validation. In the continuation of the third step, we compare the performance of D24WHFSS-based MRTPFs with extracted discriminative features by other feature selection algorithms.

In D24WHFSS, the bi-IWM accompanied by filter and wrapper methods plays the pivot role in extracting MRTPFs of $mVTTFs$ to achieve low TPT and high TPA. According to the overall summary of D24WHFSS depicted in Fig. 2, first, the filter phase is conducted per TU_x of $mVTTFs$ for ranking the transient point features (TPFs) of TU_x ($RankTPFs$ _ TU_x). Next, $RankTPFs$ _ TU_x is entered into 24WHFSS, which is repeated two times (1 D24WHFSS and 2 D24WHFSS) according to various two KFPs (See Fig. 2, sprinkler symbols), namely (RBF, RBF), (DTW, Poly) pairs. Each i D24WHFSS is categorized into four learning zones. Each zone consists of linked four-stage hybrid (L4SH) structure (SU IWSS kSVM , SU IWSS kTWSVM , SU IWSSr kSVM , and SU IWSSr kTWSVM) which led by one of them (Zone 1 led by SU IWSS kSVM , Zone 2 led by SU IWSS kTWSVM , Zone 3 led by SU IWSSr kSVM , and Zone 4 led by SU IWSSr kTWSVM). Based on the starter hybrid scenario in each zone, by permuting of the rest hybrid scenarios (three scenarios have six permutations), we have six types of L4SH structure (6L4SH structure) per zone of i D24WHFSS. Based on the chained-face of L4SH, the output of each hybrid scenario will be the input of the other scenario. After entering $RankTPFs$ _ TU_x into 6L4SH of zones (See Fig. 2, blue-face dotted line), the optimal features (OFs) obtained by L4SH 1 to L4SH 6 (x OFs 1 to x OFs 6) are recorded. Then, the intersection of zone z L4SH $^{1:6}$ -specific selected features is labeled as i D24WHFSS-based x zone z relevant TPFs (i D24WHFSS-based x Z z RTPFs). Then, the union operator is applied on i D24WHFSS-based x Z 1 RTPFs to i D24WHFSS-based x Z 4 RTPFs for selecting i D24WHFSS-based x Z $^{1:4}$ unified RTPFs (i D24WHFSS-based x Z $^{1:4}$ URTPFs) (See Fig. 2, red-face dotted line). Based on setting DKFPs on 24WHFSS, we have 1 D24WHFSS-based x Z $^{1:4}$ URTPFs and 2 D24WHFSS-based x Z $^{1:4}$ URTPFs sets. To achieve

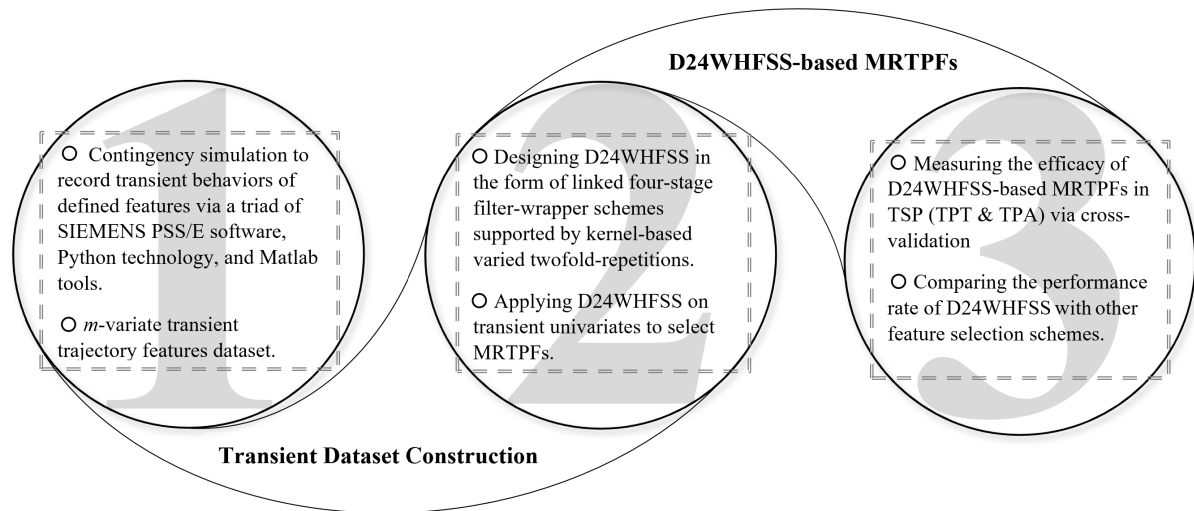


FIGURE 1. Overall workflow of FS-based TSA via D24WHFSS.

TU_x -specific MRTPFs (x MRTPFs), the union of 1 D24WHFSS-based $^xZ^{1:4}$ URTPFs and 2 D24WHFSS-based $^xZ^{1:4}$ URTPFs is calculated. Finally, the union result of 1 MRTPFs to m MRTPFs is called UMRTPFs $^{1:m}$ subset.

Besides the visual summary of D24WHFSS (See Fig. 2), for more details about the main functions of the proposed FSS, the pseudocode of D24WHFSS is shown in Table 1. According to Table 1, ‘SUCalc’ for SU-based TPFs weighting per transient univariate (TU_1 to TU_m) is situated in the main body of D24WHFSS as a primary filter-oriented function (See Table 1, Line 1-5). After ranking the weights of $TPFs_{s1}$ to $TPFs_{sl}$ related to $TPFs_{SU}^{TU_x}$ in descending order (Line 7), the necessary condition for entering each TU_x into 24WHFSS decorated in varied two-fold repetition based on DKFPs (Line 9) is provided. The $RankTPFs_{SU}^{TU_x}$ is fed to i D24WHFSS designed by the four-zone contains the linked four-stage hybrid structure (Line 11, Zone4SH). Each zone of i D24WHFSS (Line 12) is led by the starter (or leader) hybrid scenario. By execution of starter leader in each zone of i D24WHFSS (IWSS ($RankTPFs_{SU}^{TU_x}$, $^1DKFP^i$, Classifier {1}); IWSS ($RankTPFs_{SU}^{TU_x}$, $^2DKFP^i$, Classifier {2}); IWSSr ($RankTPFs_{SU}^{TU_x}$, $^1DKFP^i$, Classifier {1}); IWSSr ($RankTPFs_{SU}^{TU_x}$, $^2DKFP^i$, Classifier {2})), the optimal features of $RankTPFs_{SU}^{TU_x}$ is calculated ($^zOFs_x^{Leader}$) (Line 13). On the other hand, the permutation of the three remaining hybrid scenarios (containing IWMs of Zone4SH without starter IWM) is obtained as $[^z6L4SH_x]_{6 \times 3}$ (Line 14 and Line 15). Hence, z6L4SH_x and $^zOFs_x^{Leader}$ are considered as arguments of the ‘Link4SH’ function (Line 16) selecting six subsets of OFs ($^1:6$ OFs) per zone of i D24WHFSS. Based on the last command of ‘Link4SH’ (Line 18 of the ‘Link4SH’ function), the union result of $^1:6$ OFs is obtained as the $Z^2RTPFs_x^i$ set. By calculating $Z^2RTPFs_x^i$ per zone of i D24WHFSS, the union of $Z^1RTPFs_x^i$ to $Z^4RTPFs_x^i$ is recorded as $Z^{1:4}URTPFs_x^i$ (Line 23). Repeating the above-mentioned scenario (Line 10-23) for all KFP^i (two KFPs) cause to obtain two $Z^{1:4}URTPFs_x^i$ sets.

After conducting union operation on 1 D24WHFSS-specific URTPFs and 2 D24WHFSS-specific URTPFs ($Z^{1:4}URTPFs_x^1$ to $Z^{1:4}URTPFs_x^2$), the $Z^{1:4}MRTPFs_x^{1:2}$ (TU_x -specific MRTPFs) is obtained (Line 30). Finally, by extracting the MRTPFs per TU_x ($Z^{1:4}MRTPFs_x^{1:2}$ to $Z^{1:4}MRTPFs_m^{1:2}$), the union result of them is labeled as UMRTPFs $^{1:m}$. For a better understanding of filter and bi-IWM formulations, refer to Sections III-A to III-C.

Besides the explanations of D24WHFSS in the form of the pseudocode of its main body and various functions, we discuss the complexity of D24WHFSS in this section. The complexity of D24WHFSS is based on the bi-IWM (IWSS and IWSSr) accompanied by DSVCs (k SVM and k TWSVM). By focusing on these significant elements, we can approximate the complexity of D24WHFSS. In the worst case, the complexity of IWSS and IWSSr is $O(n)$ and $O(n^2)$, respectively [24]. Also, the complexity of SVM and TWSVM is $O(n^3)$ and $O(2 \times (n/2)^3)$, respectively [25]. Hence, the complexity of $IWSS^{kSVM/kTWSVM}$ is $O(\max\{(n \times n^3), (n \times 2 \times (n/2)^3)\})$ and $IWSSr^{kSVM/kTWSVM}$ has $O(\max\{(n^2 \times n^3), (n^2 \times 2 \times (n/2)^3)\})$ complexity. Since the complexity of the SVM is 4 times larger than of the TWSVM, the complexity of $^SUIWSS^{kSVM/kTWSVM}$ and $^SUIWSSr^{kSVM/kTWSVM}$ will be equal to $O(n \times n^3)$ and $O(n^2 \times n^3)$, respectively. On the other hand, [24] results show that the complexity of IWSSr is near to IWSS in the presence of the compacted space of optimal features in wrapper iterations. Consequently, according to the D24WHFSS scheme, D24WHFSS has $O(c \times n^4)$ complexity.

A. BI-INCREMENTAL WRAPPER MECHANISM (BI-IWM)

1) IWSS

The IWSS [27] is one of the IWM, which is used in variant forms in six types of the chained four-stage hybrid scenarios embedded in the D24WHFSS four-zone. Two approaches, including the ITPs (filter) and MLCs (wrapper), have a direct

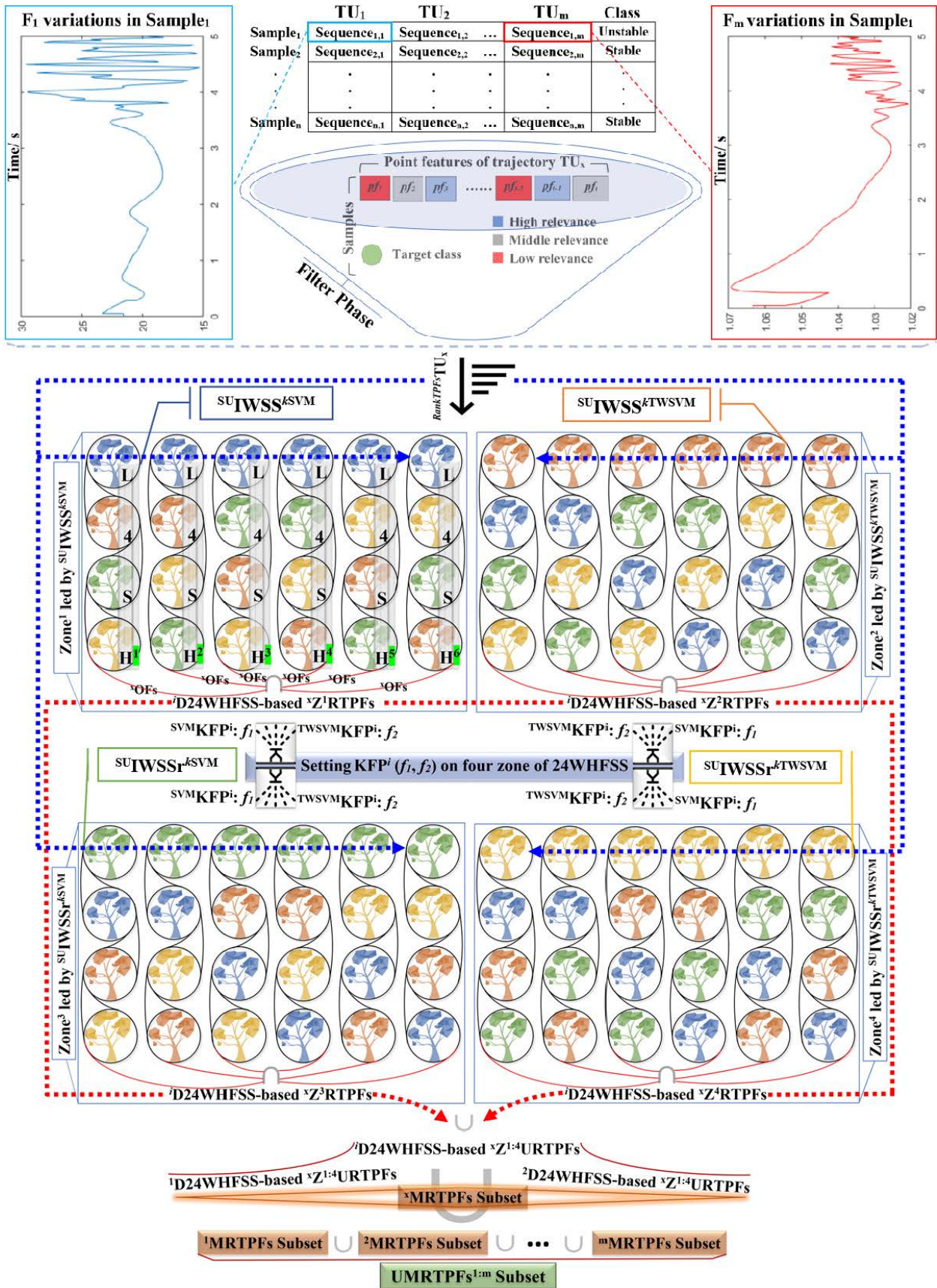


FIGURE 2. Overall process of D24WHFSS.

TABLE 1. The pseudocode of the D24WHFSS.

Main body of D24WHFSS	
Input:	Transient univariates of m -variate transient trajectory features (m VTTFs) dataset, $\{TU_x \mid x=1, 2, \dots, m\}$.
Output:	Most relevant transient point features of $TU_{1:m}$ ($1:m$ MRTPFs).
(1)	for $x=1$ to m
(2)	for $j=1$ to l // l =length ($TPFs$), $TPFs$: transient point features in TU_x .
(3)	$TPFs_j^{SU^{TU_x}} = \text{SUCalc} (TU_x^{TPFs_j});$ // SU value of $TPFs_j \in TU_x$, for more information about SU index refer to Section III. B.1
(4)	end
(5)	end
(6)	for $x=1$ to m
(7)	$\text{Rank}TPFs_{TU_x} = \text{Sort} (TPFs_{SU}^{TU_x});$ // Sort the SU-based weights of $TPFs_{SU}$ to $TPFs_l$ related to TU_x ($TPFs_{SU}^{TU_x}$) in descending order.
(8)	Classifier: {SVM, TWSVM}; DKFP: {(RBF, RBF), (DTW, Poly)}.
(9)	for $i=1$ to 2 // Exerting 24WHFSS in varied two-fold repetition based on DKFPs (dyadic 24WHFSS), $\{^i\text{D24WHFSS} \mid i=1, 2\}$.
(10)	DKFP $^i = \text{DKFP} \{i\};$
(11)	Zone4SH: $\{\text{IWSS} (\text{Rank}TPFs_{TU_x}^i, ^1\text{DKFP}^i, \text{Classifier} \{1\}), \text{IWSS} (\text{Rank}TPFs_{TU_x}^i, ^2\text{DKFP}^i, \text{Classifier} \{2\}), \text{IWSSr} (\text{Rank}TPFs_{TU_x}^i, ^1\text{DKFP}^i, \text{Classifier} \{1\}), \text{IWSSr} (\text{Rank}TPFs_{TU_x}^i, ^2\text{DKFP}^i, \text{Classifier} \{2\})\}$ // $^1\text{DKFP}^i$: the first function of DKFP plugged into SVM; $^2\text{DKFP}^i$: the second function of DKFP plugged into TWSVM.
(12)	for $z=1$ to 4 // four zone contains six type linked four-stage hybrid (6L4SH) structure per $^i\text{D24WHFSS}$.
(13)	$z_i^{OFs^{Leader}} = \text{run} (\text{Zone4SH} \{z\});$ // select starter (leader) hybrid scenario per zone of $^i\text{D24WHFSS}$, run: function execution.
(14)	restIWMs = Zone4SH-Zone4SH $\{z\};$ // contains IWMs of Zone4SH without starter IWM.
(15)	$z_i^{6L4SH_x} = \text{Permutation} ([\text{restIWMs}]);$ // $[z_i^{6L4SH_x}]_{6 \times 3}$: contains different permutations of restIWM per zone; for more information about permutation function refer to MATLAB-based 'perms' command [26].
(16)	$Z^z \text{RTPFs}_x^i = \text{Link4SH} (z_i^{6L4SH_x}, z_i^{OFs^{Leader}});$ // $Z^z \text{RTPFs}_x^i$: contain the Zone z -specific RTPFs per $^i\text{D24WHFSS}$.
(17)	if $z=1$
(18)	S = struct ('z', $Z^z \text{RTPFs}_x^i$);
(19)	else
(20)	S(end+1) = struct ('z', $Z^z \text{RTPFs}_x^i$);
(21)	end
(22)	end
(23)	$Z^{1:4} \text{URTPFs}_x^i = \text{S}(1) \cup \text{S}(2) \cup \text{S}(3) \cup \text{S}(4);$ // $Z^{1:4} \text{URTPFs}_x^i$: contain union result of Zone 1 -specific RTPFs to Zone 4 -specific RTPFs per $^i\text{D24WHFSS}$.
(24)	if $i=1$
(25)	SS = struct ('i', $Z^{1:4} \text{URTPFs}_x^i$);
(26)	else
(27)	SS(end+1) = struct ('i', $Z^{1:4} \text{URTPFs}_x^i$);
(28)	end
(29)	end
(30)	$Z^{1:4} \text{MRTPFs}_x^{1:2} = \text{SS} (1) \cup \text{SS} (2);$ // $Z^{1:4} \text{MRTPFs}_x^{1:2}$: contain union result of obtained $^1\text{D24WHFSS}$ -specific URTPFs and $^2\text{D24WHFSS}$ -specific URTPFs.
(31)	if $x=1$
(32)	SSS = struct ('x', $Z^{1:4} \text{MRTPFs}_x^{1:2}$);
(33)	else
(34)	SSS(end+1) = struct ('x', $Z^{1:4} \text{MRTPFs}_x^{1:2}$);
(35)	end
(36)	end
(37)	$\text{UMRTPFs}_{1:m} = \text{SSS} (1) \cup \text{SSS} (2) \cup \dots \cup \text{SSS} (m);$ // MRTPFs: contain union result of TU_1 -specific MRTPFs to TU_m -specific MRTPFs.
Function: IWSSr (A, B, C)	
(1)	Sel=A {1}; // the first element of A array (feature with highest SU) insert in Sel.
(2)	Learning model: {B plugged in C}.
(3)	AccSel=PerEval (Learning Model, $[data]_{n \times \text{Sel with } L}$); // L: [class label] $_{n \times 1}$; n: number of sample, PerEval: Performance Evaluation.
(4)	for $i=2$ to length (A)
(5)	OptFea = \emptyset ;
(6)	for $r=1$ to length (Sel)
(7)	Sel $^{temp} = \text{update}_{\text{SelSub}} (\text{copy}(\text{Sel}), \text{swap}(\text{Sel}\{r\}, A\{i\}));$
(8)	AccSel $^{temp} = \text{PerEval} (\text{Learning Model}, [data]_{n \times \text{Sel}^{temp} \text{ with } L});$

TABLE 1. (Continued.) The pseudocode of the D24WHFSS.

```

(9)  if (AccSeltemp>AccSel)
(10)  OptFea=swap (Sel{r}, A {i});
(11)  Acc= AccSeltemp;
(12)  end
(13)  end
(14)  Seltemp=updateSelSub (copy(Sel), add (A {i}));
(15)  AccSeltemp=PerEval (Learning model, [data]n×Seltemp with L);
(16)  if (AccSeltemp>AccSel)
(17)  OptFea=Seltemp;
(18)  AccSel=AccSeltemp;
(19)  end
(20)  if (OptFea != null)
(21)  update (Sel, OptFea);
(22)  end
(23) end
(24) return Sel;

```

Function: IWSS (D, E, F)

```

(1) Sel=D {1};
(2) Learning model: {E plugged in F}.
(3) AccSel=PerEval (Learning Model, [data]n×Sel with L);
(4) for i=2 to length (D)
(5)  Seltemp=add (copy(Sel), D {i});
(6)  AccSeltemp=PerEval (Learning Model, [data]n×Seltemp with L);
(7)  if (AccSeltemp>AccSel)
(8)  add (Sel, D {i});
(9)  AccSel=AccSeltemp;
(10) end
(11) end
(12) return Sel;

```

Function: Link4SH (G, H)

```

(1) for s=1 to 6 // exerting the six types of L4SH according to the number of permutation of restIWM without leader hybrid scenario
(2)  perIWM=G (s, :);
(3)  ROFsstage1 =Sort (H);
(4)  perIWM(1).{1} = ROFsstage1 ;
(5)  OFsstage2 =run (perIWM(1));
(6)  ROFsstage2 =Sort ( OFsstage2 );
(7)  perIWM(2).{1} = ROFsstage2 ;
(8)  OFsstage3 =run (perIWM(2));
(9)  ROFsstage3 =Sort ( OFsstage3 );
(10) perIWM(3).{1} = ROFsstage3 ;
(11) OFsstage4 =run (perIWM(3));
(12) if s=1
(13)  OFsstagestotal =struct('s', OFsstage4);
(14) else
(15)  OFsstagestotal (end+1) =struct('s', OFsstage4);
(16) end
(17) end
(18) RTPFs = ∩ [ OFsstagestotal (1) : OFsstagestotal (6)].
(19) return RTPFs ;

```

* The pseudocode of the D24WHFSS is rewritten based on MATLAB commands in the MATLAB environment.

relationship in the growth of IWSS tree branches. First, features ranking in descending order based on information theory-based indices (ITIs) is performed to determine how features are entered into the IWSS tree. Then, the first branch of the IWSS tree grows by training the MLC via the feature

with the highest ITI (f_{h1}), and the prediction accuracy in the presence of f_{h1} ($Acc(f_{h1})$) is obtained. The subsequent growth is related to the participation of the second-rank feature (f_{h2}) with f_{h1} to train MLC, and the obtained result is labeled as $Acc(f_{h1}, f_{h2})$. If the comparison of $Acc(f_{h1}, f_{h2})$ and $Acc(f_{h1})$

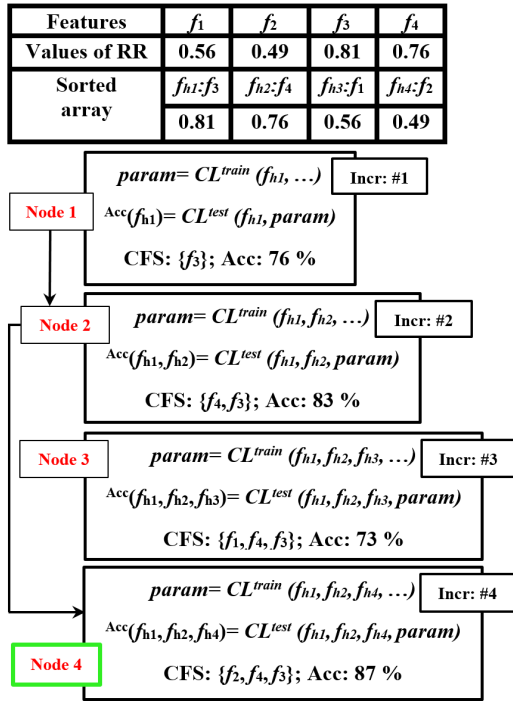


FIGURE 3. IWSS algorithm [22].

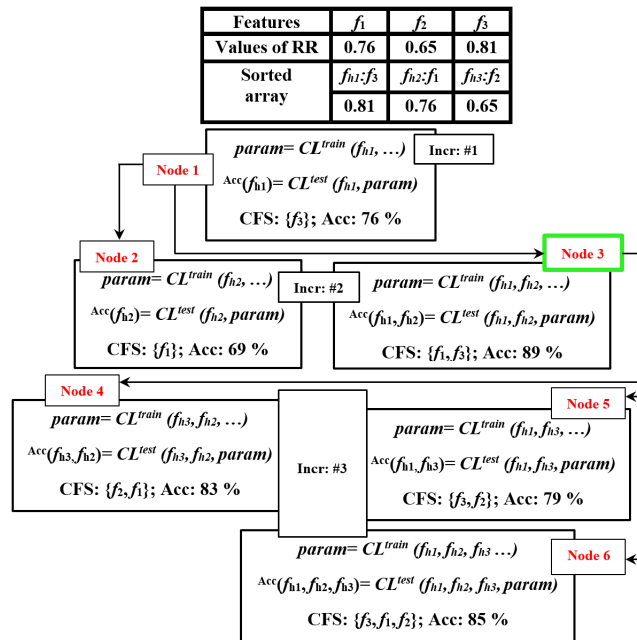


FIGURE 4. IWSSr algorithm [22].

reports superiority of coupling f_{h1} and f_{h2} , the f_{h3} added to (f_{h1}, f_{h2}) as 3rd element for MLC training. On the other hand, if the $\text{Acc}(f_{h1}, f_{h2})$ is lower than $\text{Acc}(f_{h1})$, f_{h2} is discarded from the candidate file, and a couple of f_{h3} and f_{h1} are used for MLC training. An example of how the IWSS algorithm works is shown in Fig. 3.

2) IWSSr

Another IWM accompanied by IWSS for selecting optimal features in D24WHFSS is the IWSSr algorithm [24].

According to the dual basic requirements to form the IWSS tree (ITPs and MLCs), the IWSSr is no exception in this matter. The IWSSr works differently than IWSS in the tree branches grow. After sorting features based on ITIs, f_{h1} is placed in the candidate file as the first element and used for training the MLC. After obtaining $\text{Acc}(f_{h1})$, two branches develop from the f_{h1} -based node. First, f_{h1} is replaced with f_{h2} (See Fig. 4, Node 2), and MLC is trained by f_{h2} . Second, couple f_{h1} and f_{h2} (See Fig. 4, Node 3) participate in MLC training. According to Fig. 4, the obtained results is manifested that among Node 1 ($\text{Acc}(f_{h1})$), Node 2 ($\text{Acc}(f_{h2})$), and Node 3 ($\text{Acc}(f_{h1}, f_{h2})$), Node 3 is selected for subsequent increment. Node 3 growth via f_{h3} cause to create the Node 4 with (f_{h2}, f_{h3}) -based MLC, Node 5 with (f_{h1}, f_{h3}) -based MLC, and (f_{h1}, f_{h2}, f_{h3}) -based MLC. Such an increment under Node 3 does not improve the prediction accuracy, and consequently, (f_{h1}, f_{h2}) of Node 3 is introduced as optimal features.

B. FILTER AND WRAPPER METHODS PLUGGED IN BI-IWM of D24WHFSS

1) SU-BASED FILTER IN BI-IWM

The SU index [28] as the symmetrical measure is considered a preliminary step of bi-IWM to specify the importance degree of features. The SU via interlacing basic ITIs measures the amount of feature relevance with the target class. Based on triple basic ITIs, namely entropy, conditional entropy, and mutual information (MI), the SU index is defined as:

$$SU(TU_x^{TPFs_j}, L) = 2 \frac{MI(TU_x^{TPFs_j}, L)}{H(TU_x^{TPFs_j}) + H(L)} \quad (1)$$

where $TU_x^{TPFs_j}$ indicates j^{th} transient point features of TU_x , and L as class label reports the status of transient cases. According to (1), SU supported by the entropy index, which is defined as:

$$H(K) = - \sum_{k \in K} p(k) \log p(k) \quad (2)$$

In (2), based on the probability density function $p(k) = \text{Pr}\{K = k\}$, K 's entropy (K represent discrete random variable) is calculated. Another index situated in (1) is mutual information (MI) which is given by:

$$MI(TU_x^{TPFs_j}, L) = H(TU_x^{TPFs_j}) - H(TU_x^{TPFs_j} | L) \quad (3)$$

In (3), $H(TU_x^{TPFs_j} | L)$ reflects the conditional entropy and is defined as:

$$H(TU_x^{TPFs_j} | L) = - \sum_{f \in TU_x^{TPFs_j}} \sum_{l \in L} p(f, l) \log p(f | l) \quad (4)$$

2) WRAPPER METHODS IN BI-IWM

a: SVM CLASSIFIER

SVM [29] is one of the most popular hyperplane-based classifiers that focus on plotting the optimal separating hyperplane between binary or multi-label considering triple

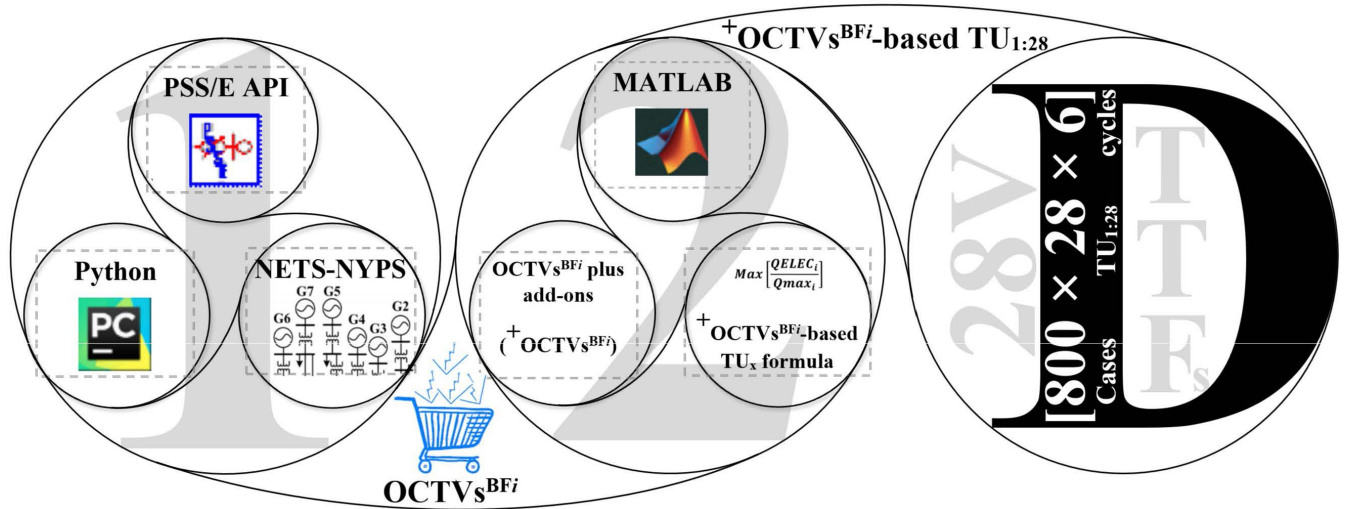


FIGURE 5. Two-step transient data creation mechanism (2STDCM).

fundamental principles (3FPs), namely margin maximization, structural-risk minimization, and avoiding overfitting. Besides the soft or hard margin-based idea for linear classification, the kernel-based approach paves the way for finding the zero-error nonlinear decision boundary on non-separable HDTS. Plugging kernel into SVM computations causes projecting data into separable space. Hence, SVM can be reformulated by exploiting the kernel trick as follow:

$$\begin{aligned}
 a^* &= \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{k=1}^l \alpha_k; \\
 0 &\leq \alpha_i \leq C, \quad \sum_{j=1}^l \alpha_j y_j = 0, \quad i, j = 1, \dots, l \quad (5)
 \end{aligned}$$

In this paper, two types of efficient kernels are used as substitutes for $K(x_i, x_j)$: 1) radial basis function (RBF) [29] and 2) dynamic time warping in RBF (DTW RBF) [30]. The concise explanations of RBF and DTW RBF kernels are as follow:

(1) RBF kernel: The RBF provides point-to-point matching for pattern discovery in feature space. The non-elastic RBF kernel is defined as:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (6)$$

$\|x - x'\|^2$ in (6) represents the squared Euclidean distance for calculating the distance between two data points.

(2) DTW RBF kernel: Changing the RBF kernel formula by replacing RBF's distance function with DTW causes defining elastic kernel, which brings the nonlinear pattern matching on feature space. The DTW distance is given by:

$$\begin{aligned}
 &\text{distance}^{DTW}(A_1^p, B_1^q) \\
 &= d(a(p), b(q)) + \text{Min} \begin{pmatrix} \text{distance}^{DTW}(A_1^{p-1}, B_1^q) \\ \text{distance}^{DTW}(A_1^p, B_1^{q-1}) \\ \text{distance}^{DTW}(A_1^p, B_1^{q-1}) \end{pmatrix} \quad (7)
 \end{aligned}$$

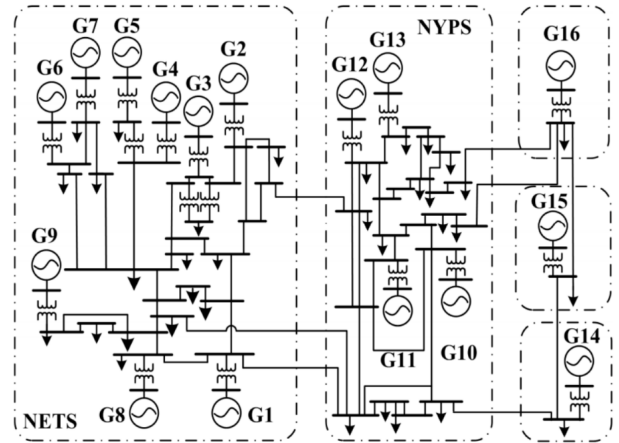


FIGURE 6. Single line diagram of NETS-NYPS test system.

Based on DTW distance, the DTW RBF kernel is defined as (8):

$$K(x, x') = \exp\left(-\frac{[\text{distance}^{DTW}(A_1^p, B_1^q)]^2}{2\sigma^2}\right) \quad (8)$$

Finally, solving (9) leads to drawing 3FPs-based separating hyperplane in HDTS:

$$\begin{aligned}
 f(x) &= \text{sgn}\left(\sum_{i \in s} \alpha_i y_i K(x_i, x) + b\right); \\
 b &= \frac{1}{s} \sum_{i \in s} \left[y_i - \sum_j \alpha_j y_j K(x_j, x_i) \right] \quad (9)
 \end{aligned}$$

b: TWSVM CLASSIFIER

Making the changes in the mathematical structure of the SVM learning model caused the offering of a

TABLE 2. 28-variate transient trajectory features (TU_{1:28}).

Math formula
$TU_1^{t_m} = \text{Max}([\frac{PELEC_i}{P_{\text{max}_i}}]^{i=1:N_{\text{genbus}}})$
$TU_2^{t_m} = \text{Var}([\frac{PELEC_i}{P_{\text{max}_i}}]^{i=1:N_{\text{genbus}}})$
$TU_3^{t_m} = \text{Max}([\frac{QELEC_i}{Q_{\text{max}_i}}]^{i=1:N_{\text{genbus}}})$
$TU_4^{t_m} = \text{Min}([\frac{QELEC_i}{Q_{\text{max}_i}}]^{i=1:N_{\text{genbus}}})$
$TU_5^{t_m} = \text{Var}([\frac{QELEC_i}{Q_{\text{max}_i}}]^{i=1:N_{\text{genbus}}})$
$TU_6^{t_m} = \text{Max}([VOLT_i]^{i=1:N_{\text{bus}}})$
$TU_7^{t_m} = \text{Var}([VOLT_i]^{i=1:N_{\text{bus}}})$
$TU_8^{t_m} = \text{Max}([VANGLE_i]^{i=1:N_{\text{bus}}}); \text{ slack bus} = 0$
$TU_9^{t_m} = \text{Min}([VANGLE_i]^{i=1:N_{\text{bus}}}); \text{ slack bus} = 0$
$TU_{10}^{t_m} = \text{Var}([VANGLE_i]^{i=1:N_{\text{bus}}}); \text{ slack bus} = 0$
$TU_{11}^{t_m} = \text{Max}(\text{abs}([VANGLE_i - VANGLE_j]^{i,j=1:N_{\text{bus}}}))$
$TU_{12}^{t_m} = \text{Mean}(\text{abs}([VANGLE_i - VANGLE_j]^{i,j=1:N_{\text{bus}}}))$
$TU_{13}^{t_m} = \text{Var}(\text{abs}([VANGLE_i - VANGLE_j]^{i,j=1:N_{\text{bus}}}))$
$TU_{14}^{t_m} = \frac{\sum_{i=1}^{N_{\text{busgen}}} QLOAD_i}{\sum_{i=1}^{N_{\text{busgen}}} QELEC_i}$
$TU_{15:28}^{t_m} = \text{Gradient of } TU_1 \text{ to } TU_{14}$

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, Q_{max}= maximum amount of machine reactive power, Q_{load}= 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.

cross hyperplane-based classifier called generalized proximal eigenvalue support vector machine (GEPSSVM) [31]. In GEPSSVM, finding two hyperplanes that each hyperplane takes the nearest distance from the samples of a class and

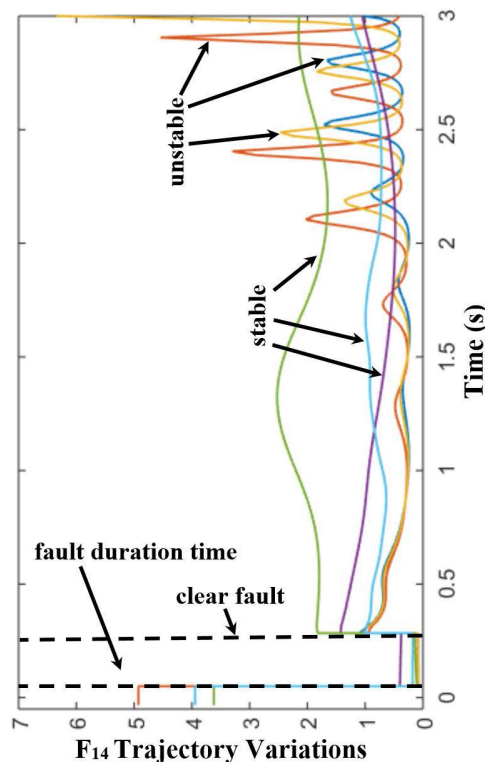


FIGURE 7. Stable and unstable samples based on F₁₄ variations [22].

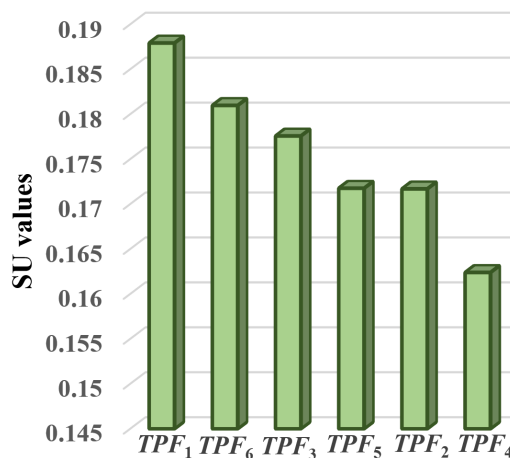


FIGURE 8. SU amount of TU₁₅^{TPFs1:6}.

the farthest distance from the samples of another one is on the agenda. Achieving the high-performance classification based on GEPSSVM motivates DM scholars to design a novel classifier based on the GEPSSVM principle. TWSVM [32] is the name of the GEPSSVM-based classification model that regards the new formulation for plotting the separating cross hyperplanes. The TWSVM-based optimization problems are as follows:

$$\min_{w_1, b_1, q} \frac{1}{2} \|Pw_1 + e_1 b_1\|^2 + c_1 e_2^T q$$

$$s.t. -(Qw_1 + e_2b_1) + q \geq e_2, q \geq 0 \quad (10)$$

$$\min_{w_2, b_2, q} \frac{1}{2} \|Qw_2 + e_2b_2\|^2 + c_2e_1^T q$$

$$s.t. (Pw_2 + e_1b_2) + q \geq e_1, q \geq 0 \quad (11)$$

where $c_1, c_2 > 0$ are parameters, and e_1 and e_2 are vectors of ones with proper dimensions. By obtaining the Karush–Kuhn–Tucker (KKT) conditions for (10) and (11) based on applying the Lagrangian function on (10) and (11); and also combining some relations, the dual optimization problem of (10) and (11) are obtained as follows:

$$dual\ TWSVM^1 : \max\{e_2^T \alpha - \frac{1}{2} \alpha^T G(H^T H)^{-1} G^T \alpha\} \quad (12)$$

$$dual\ TWSVM^2 : \max\{e_1^T \psi - \frac{1}{2} \psi^T P(Q^T Q)^{-1} P^T \psi\} \quad (13)$$

Based on quadratic programming applied on (12) and (13), α and ψ are obtained. Hence, necessary conditions to take the values of $[w^{(1)}, b^{(1)}]$ and $[w^{(2)}, b^{(2)}]$ for drawing the cross hyperplanes on a binary classification problem is provided:

$$X^T w^{(1)} + b^{(1)} = 0 \text{ and } X^T w^{(2)} + b^{(2)} = 0 \quad (14)$$

Finally, predicting the class label of an unseen case is given by:

$$Class\ x = \arg_v \min |x^T w^{(v)} + b^{(v)}|; \quad v = 1, 2 \quad (15)$$

To achieve high-performance classification in the non-separable HDTs, empowering TWSVM by embedding the kernel functions on TWSVM computations is the best solution [32]. Hence, we have:

$$K(x^T, C^T)u^{(1)} + b^{(1)} = 0 \text{ and } K(x^T, C^T)u^{(2)} + b^{(2)} = 0 \quad (16)$$

In (16), $C^T = [A\ B]^T$ and K indicate the kernel function. Solving the (17) and (18) leads to obtaining the $[u^{(1)}\ b^{(1)}]^T$ and $[u^{(2)}\ b^{(2)}]^T$ vectors.

$$K\ TWSVM^1 : \min_{u^{(1)}, b^{(1)}, q} \frac{1}{2} \|(K(A, C^T)u^{(1)} + e_1b^{(1)})\|^2 + c_1e_2^T q$$

$$s.t. -(K(B, C^T)u^{(1)} + e_2b^{(1)}) + q \geq e_2,$$

$$q \geq 0 \quad (17)$$

$$K\ TWSVM^2 : \min_{u^{(2)}, b^{(2)}, q} \frac{1}{2} \|(K(B, C^T)u^{(2)} + e_2b^{(2)})\|^2 + c_2e_1^T q$$

$$s.t. (K(A, C^T)u^{(2)} + e_1b^{(2)}) + q \geq e_1,$$

$$q \geq 0 \quad (18)$$

In (17) and (18), the K is replaced with RBF (discussed in (a) section of III. B. 2) and the polynomial (Poly) [33] kernels. The following definition is related to the Poly:

Poly kernel: In linear kernel relation [33], by setting degree (d) to greater than one, the Poly kernel is defined as follows:

$$K(x, x') = (\alpha x^T x' + c)^d \quad (19)$$

TABLE 3. Different uncertainties to generate dynamic responses [23], [34].

Parameter	Parameter elements
convert the constant MVA load for a specified grouping of network loads to a specified mixture of the constant MVA, constant current, and constant admittance load characteristics	^a loadin [1] is the percent of active power load to be converted to the constant current characteristic.
	loadin [2] is the percent of active power load to be converted to the constant admittance characteristic.
	loadin [3] is the percent of reactive power load to be converted to the constant current characteristic.
	loadin [4] is the percent of reactive power load to be converted to the constant admittance characteristic.
Type of disturbance	Outage [substation, generator, line outages]
Fault duration time	0.23 (second) ^b

^aloadin: an array of four elements, the value of each element is percent of the load being converted.

IV. EXPERIMENTAL DESIGN

A. CREATING TRANSIENT DATASET

For FS-based TSA, creating the transient dataset is the preliminary task of the three-step proposed framework in this paper (See Fig. 1, Step 1). In this regard, we design the two-step transient data creation mechanism (2STDCM) as shown in Fig. 5. In 2STDCM, first, output channel transient values per basic features ($OCTVs^{BF_i}$) are recorded. The BF_i includes the bus voltages (VOLT), voltage phase angle (VANGLE), machine active power (PELEC), machine reactive power (QELEC), and reactive power consumption (QLOAD)). The $OCTVs^{BF_i}$ is obtained via Python-based contingency simulation (PCS) supported by the application program interface (API) functions of the SIEMENS power system simulator for engineering (PSS/E) [34]. For more information about Python scripting for dynamic simulation based on PSS/E API ('psspy' module) refers to Table 4. The contingency simulation is conducted on the New England test system- New York power system (NETS-NYPS) (See Fig. 6) [35]. The transient cases are stemmed from substation outages, generator outages, and line outages by setting disturbance different parameters (fault duration time: 0.23 seconds with 0.0167 seconds time step and the fault clearing time is set after the end of fault duration time). For gathering severe transient samples, the uncertainty factor is considered by the convert load (CONL) API of PSS/E situated in PCS that causes the setting of different load characteristics for converting active and reactive power load (See Table 3; first row) [23], [34]. The second step of 2STDCM includes a Matlab-based script for adding the required add-ons to $OCTVs^{BF_i}$ (e.g., geometric functions and other electrical parameters) called $+OCTVs^{BF_i}$. Finally, transient responses of 28-variate transient trajectory features (28VTTFs) enumerated in Table 2 [36], [37], and [38] are obtained. Based on performing 2STDCM, we have transient dataset containing 800 (No. transient cases) \times 28 (TU_{1.28}) \times 6 (No. observed cycles). For example, some stable and unstable excursions regarding TU₁₄ trajectory (The proportion of total QLOAD to total QELEC) are shown in Fig. 7. We executed

TABLE 4. Python scripting for dynamic simulation based on PSS/E API ('PSSPY Module') [37].

```

(1) for k in xrange(1,buscon+1):
(2)   x=busm[0,k-1]
(3)   psspy.pssinit(15000000) #Initialize PSS/E, requested bus size
(4)   psspy.case(CASE) #PSS/E Saved Case file and transfers its data into the PSS/E working case
(5)   report = r"C:\Users\...\report"+k+'.txt'
(6)   ierr=psspy.progress_output(2,report,[0,0]) #specify the progress output device; direct output to a file and open with
      carriage control format and, for files, for overwrite of existing files
(7)   psspy.conl(-1,1,1) #initialize for load conversion
(8)   psspy.conl(-1,1,2,[0,0],[100,0,0,100]) #convert the constant MVA load for a specified grouping of network loads to a
      specified mixture of the constant MVA, constant current, and constant admittance load characteristics
(9)   psspy.conl(-1,1,3) # postprocessing housekeeping
(10)  psspy.cong() #convert generators from their power flow representation in preparation for switching studies and dynamic
      simulations
(11)  psspy.ordr() #calculate a sparsity preserving ordering of buses in preparation for the processing of network matrices
(12)  psspy.fact() #factorize the network admittance matrix in preparation for switching studies and dynamic simulations
(13)  psspy.tysl() #run switching study network solutions
(14)  case_root=os.path.splitext(CASE)[0]
(15)  psspy.save(case_root+"_C"+k+'.sav') #save the PSS/E working case in a Saved Case File
(16)  psspy.dyre_new(dyrefile="C:\Program Files (x86)\PTI\PSSE33\EXAMPLE\NETS\NYP5 68 Bus Syste.dyr") #read a
      dynamics data file, and place the model references specified on its data records into dynamics working memory
(17)  psspy.chsb(sid=0,all=1,status=[-1,-1,-1,1,3,0]) #chsb: specify the simulation variables to monitor during dynamic
      simulation. run (activity chsb); machine reactive power: QELEC
(18)  psspy.chsb(sid=0,all=1,status=[-1,-1,-1,1,26,0]) #chsb: specify the simulation variables to monitor during dynamic
      simulation. run (activity chsb); reactive power consumption: Qload
(19)  psspy.snap(sfile="C:\Program Files (x86)\PTI\PSSE33\EXAMPLE\python_test_"+k+'.snp') #save PSS/E dynamics
      working memory into a Snapshot File
(20)  psspy.dynamics_solution_param_2(intgar=[11,271,0,0,0,0,1],realar=[1,0.0005,0,0167]) #modify the dynamic simulation
      solution parameters in dynamics working memory, e.g.; intgar [1] for network solution maximum number of iterations and
      intgar [2] for number of output channels being monitored
(21)  psspy.strt(outfile="C:\Users\...\out\python_test_"+k+'.out') #recorded during the dynamic simulation
(22)  psspy.throwPssExceptions = True #throw exceptions instead of returning error codes values
(23)  psspy.run(tpause=0.03677) #fault occurrence time
(24)  psspy.dist_bus_fault(ibus=x,units=3,values=[0,0]) #apply a fault at a bus during dynamic sim; ibus: # bus which the fault
      is to be placed; basekv: base voltage in kV used to calculate the per unit fault admittance.
(25)  psspy.run(tpause=0.2677) #fault clearing time
(26)  psspy.dist_clear_fault() #clear a fault during dynamic simulation
(27)  psspy.dist_bus_trip(ibus=x) #disconnect a bus during dynamic simulation
(28)  psspy.run(tpause=1.1027) #simulation time termination

```

TABLE 5. The SU-based ranking of transient point features of TU_{1:28}.

Input	Rank ^{TPFs} TU _x based on	Input	Rank ^{TPFs} TU _x based on SU
TU ₁	{ ¹ TPF ₁ , ¹ TPF ₆ , ¹ TPF ₃ , ¹ TPF ₅ , ¹ TPF ₂ , ¹ TPF ₄ }	TU ₁₅	{ ¹⁵ TPF ₁ , ¹⁵ TPF ₆ , ¹⁵ TPF ₃ , ¹⁵ TPF ₅ , ¹⁵ TPF ₂ , ¹⁵ TPF ₄ }
TU ₂	{ ² TPF ₁ , ² TPF ₂ , ² TPF ₃ , ² TPF ₄ , ² TPF ₅ , ² TPF ₆ }	TU ₁₆	{ ¹⁶ TPF ₁ , ¹⁶ TPF ₂ , ¹⁶ TPF ₃ , ¹⁶ TPF ₄ , ¹⁶ TPF ₅ , ¹⁶ TPF ₆ }
TU ₃	{ ³ TPF ₅ , ³ TPF ₁ , ³ TPF ₃ , ³ TPF ₄ , ³ TPF ₆ , ³ TPF ₂ }	TU ₁₇	{ ¹⁷ TPF ₂ , ¹⁷ TPF ₄ , ¹⁷ TPF ₃ , ¹⁷ TPF ₁ , ¹⁷ TPF ₅ , ¹⁷ TPF ₆ }
TU ₄	{ ⁴ TPF ₄ , ⁴ TPF ₁ , ⁴ TPF ₃ , ⁴ TPF ₅ , ⁴ TPF ₆ , ⁴ TPF ₂ }	TU ₁₈	{ ¹⁸ TPF ₁ , ¹⁸ TPF ₃ , ¹⁸ TPF ₆ , ¹⁸ TPF ₅ , ¹⁸ TPF ₄ , ¹⁸ TPF ₂ }
TU ₅	{ ⁵ TPF ₁ , ⁵ TPF ₂ , ⁵ TPF ₃ , ⁵ TPF ₄ , ⁵ TPF ₅ , ⁵ TPF ₆ }	TU ₁₉	{ ¹⁹ TPF ₁ , ¹⁹ TPF ₂ , ¹⁹ TPF ₃ , ¹⁹ TPF ₄ , ¹⁹ TPF ₅ , ¹⁹ TPF ₆ }
TU ₆	{ ⁶ TPF ₃ , ⁶ TPF ₂ , ⁶ TPF ₄ , ⁶ TPF ₆ , ⁶ TPF ₁ , ⁶ TPF ₅ }	TU ₂₀	{ ²⁰ TPF ₅ , ²⁰ TPF ₃ , ²⁰ TPF ₆ , ²⁰ TPF ₄ , ²⁰ TPF ₂ , ²⁰ TPF ₁ }
TU ₇	{ ⁷ TPF ₁ , ⁷ TPF ₂ , ⁷ TPF ₃ , ⁷ TPF ₄ , ⁷ TPF ₅ , ⁷ TPF ₆ }	TU ₂₁	{ ²¹ TPF ₁ , ²¹ TPF ₂ , ²¹ TPF ₃ , ²¹ TPF ₄ , ²¹ TPF ₅ , ²¹ TPF ₆ }
TU ₈	{ ⁸ TPF ₂ , ⁸ TPF ₅ , ⁸ TPF ₃ , ⁸ TPF ₁ , ⁸ TPF ₄ , ⁸ TPF ₆ }	TU ₂₂	{ ²² TPF ₅ , ²² TPF ₄ , ²² TPF ₁ , ²² TPF ₆ , ²² TPF ₃ , ²² TPF ₂ }
TU ₉	{ ⁹ TPF ₂ , ⁹ TPF ₁ , ⁹ TPF ₃ , ⁹ TPF ₅ , ⁹ TPF ₆ , ⁹ TPF ₄ }	TU ₂₃	{ ²³ TPF ₆ , ²³ TPF ₃ , ²³ TPF ₅ , ²³ TPF ₄ , ²³ TPF ₂ , ²³ TPF ₁ }
TU ₁₀	{ ¹⁰ TPF ₁ , ¹⁰ TPF ₂ , ¹⁰ TPF ₃ , ¹⁰ TPF ₄ , ¹⁰ TPF ₅ , ¹⁰ TPF ₆ }	TU ₂₄	{ ²⁴ TPF ₁ , ²⁴ TPF ₂ , ²⁴ TPF ₃ , ²⁴ TPF ₄ , ²⁴ TPF ₅ , ²⁴ TPF ₆ }
TU ₁₁	{ ¹¹ TPF ₃ , ¹¹ TPF ₁ , ¹¹ TPF ₂ , ¹¹ TPF ₄ , ¹¹ TPF ₆ , ¹¹ TPF ₅ }	TU ₂₅	{ ²⁵ TPF ₆ , ²⁵ TPF ₃ , ²⁵ TPF ₅ , ²⁵ TPF ₂ , ²⁵ TPF ₄ , ²⁵ TPF ₁ }
TU ₁₂	{ ¹² TPF ₁ , ¹² TPF ₂ , ¹² TPF ₃ , ¹² TPF ₄ , ¹² TPF ₅ , ¹² TPF ₆ }	TU ₂₆	{ ²⁶ TPF ₁ , ²⁶ TPF ₂ , ²⁶ TPF ₃ , ²⁶ TPF ₄ , ²⁶ TPF ₅ , ²⁶ TPF ₆ }
TU ₁₃	{ ¹³ TPF ₁ , ¹³ TPF ₂ , ¹³ TPF ₃ , ¹³ TPF ₄ , ¹³ TPF ₅ , ¹³ TPF ₆ }	TU ₂₇	{ ²⁷ TPF ₁ , ²⁷ TPF ₂ , ²⁷ TPF ₃ , ²⁷ TPF ₄ , ²⁷ TPF ₅ , ²⁷ TPF ₆ }
TU ₁₄	{ ¹⁴ TPF ₁ , ¹⁴ TPF ₂ , ¹⁴ TPF ₃ , ¹⁴ TPF ₄ , ¹⁴ TPF ₅ , ¹⁴ TPF ₆ }	TU ₂₈	{ ²⁸ TPF ₁ , ²⁸ TPF ₂ , ²⁸ TPF ₃ , ²⁸ TPF ₄ , ²⁸ TPF ₅ , ²⁸ TPF ₆ }

Symbol: TPF: transient point feature; TU_x: ^xth transient univariate; ^xTPF_j: ^jth TPFs of TU_x; Rank^{TPFs}TU_x: Sorted TPFs of TU_x based on SU

all the simulations on a programming platform employing a Surface Pro 4 with an Intel Core i5-6300 2.5 GHz processor and 4 GB RAM.

B. SELECTING UMRTPFs^{1:28} SET

In this section, we elaborate on how to select the ^{1:28}MRTPFs set containing the union result of TU₁-specific MRTPFs to TU₂₈-specific MRTPFs based on performing D24WHFSS

on 28VTTFs (TU_{1:28}). According to Fig. 2, in the first step of D24WHFSS, the SU-based filter is applied on 28VTTFs for ranking the importance degree of the TPFs of TU_{1:28} (Rank^{TPFs}TU₁ to Rank^{TPFs}TU₂₈). According to descriptions of the filter method in Section III.B.1 (See (1) to (4)), the SU values of transient point features (TPF_{s_j}) per TU_x are obtained (TPF_{s_j}SU^{TU_x}). As an illustration of the SU amounts related to the six TPFs (6 cycles observed of TU₁₅(TU₁₅^{TPF5:6}),

TABLE 6. Selecting ${}^1D24WHFSS\text{-}2Z^{1:4}RTPFs$ from ${}^{RankTPFs}TU_2$ Via KPF^1 -Based $D24WHFSS$ (1D24WHFSS).

${}^1D24WHFSS\text{-}2Z^1RTPFs$					
OFs-L4SH ¹	OFs-L4SH ²	OFs-L4SH ³	OFs-L4SH ⁴	OFs-L4SH ⁵	OFs-L4SH ⁶
$\{{}^2TPF_1, {}^2TPF_3\}$	$\{{}^2TPF_1, {}^2TPF_3\}$	$\{{}^2TPF_1, {}^2TPF_3\}$	$\{{}^2TPF_1, {}^2TPF_4\}$	$\{{}^2TPF_1, {}^2TPF_4\}$	$\{{}^2TPF_1, {}^2TPF_4\}$
${}^1D24WHFSS\text{-}2Z^1RTPFs = \cap [OFs\text{-}L4SH^h]_{h=1:6}$					
$\{{}^2TPF_1\}$					
${}^1D24WHFSS\text{-}2Z^2RTPFs$					
OFs-L4SH ¹	OFs-L4SH ²	OFs-L4SH ³	OFs-L4SH ⁴	OFs-L4SH ⁵	OFs-L4SH ⁶
$\{{}^2TPF_1, {}^2TPF_4\}$	$\{{}^2TPF_1, {}^2TPF_4\}$	$\{{}^2TPF_1, {}^2TPF_4\}$	$\{{}^2TPF_1, {}^2TPF_4\}$	$\{{}^2TPF_1, {}^2TPF_4\}$	$\{{}^2TPF_1, {}^2TPF_4\}$
${}^1D24WHFSS\text{-}2Z^2RTPFs = \cap [OFs\text{-}L4SH^h]_{h=1:6}$					
$\{{}^2TPF_1, {}^2TPF_4\}$					
${}^1D24WHFSS\text{-}2Z^3RTPFs$					
OFs-L4SH ¹	OFs-L4SH ²	OFs-L4SH ³	OFs-L4SH ⁴	OFs-L4SH ⁵	OFs-L4SH ⁶
$\{{}^2TPF_1, {}^2TPF_3\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$
$\{{}^2TPF_1, {}^2TPF_5\}$					
${}^1D24WHFSS\text{-}2Z^4RTPFs$					
OFs-L4SH ¹	OFs-L4SH ²	OFs-L4SH ³	OFs-L4SH ⁴	OFs-L4SH ⁵	OFs-L4SH ⁶
$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$	$\{{}^2TPF_1, {}^2TPF_5\}$
${}^1D24WHFSS\text{-}2Z^4RTPFs = \cap [OFs\text{-}L4SH^h]_{h=1:6}$					
$\{{}^2TPF_1, {}^2TPF_5\}$					
${}^1D24WHFSS\text{-}2Z^{1:4}URTPFs = \cup [{}^1D24WHFSS\text{-}2Z^iRTPFs]_{i=1:4}$					
$\{{}^2TPF_1, {}^2TPF_4, {}^2TPF_5\}$					

Symbol: TPF: transient point feature; TU_x : x^{th} transient univariate; iTPF_j : j^{th} TPFs of TU_x ; ${}^{RankTPFs}TU_x$: Sorted TPFs of TU_x , based on SU; ${}^1D24WHFSS\text{-}2Z^iRTPFs$: Selecting six OFs set based on performing six type linked four-stage hybrid (6L4SH: L4SH¹ to L4SH⁶) of 1D24WHFSS per Zone; z: Zone 1 to 4.

refer to Fig. 8. According to Fig. 8, among the TPFs of TU_{15} , $TU_{15}^{TPFs_{s1}}$ and $TU_{15}^{TPFs_{s4}}$ get the first (high SU) and last (low SU) rank, respectively. Then, by sorting $TU_{15}^{TPFs_{s1:6}}$ in descending manner, ${}^{RankTPFs}TU_{15}$ is entered into the bi-IWM of D24WHFSS. The sorted face of $TU_{15}^{TPFs_{s1:6}}$ to $TU_{28}^{TPFs_{s1:6}}$ (${}^{RankTPFs}TU_{15}$ to ${}^{RankTPFs}TU_{28}$) based on SU is shown in Table 5.

After ranking the TPFs of $TU_{1:28}$ based on SU values (${}^{RankTPFs}TU_{15}$ to ${}^{RankTPFs}TU_{28}$), the ${}^{RankTPFs}TU_{1:28}$ are entered into the varied bi-IWM (IWSS and IWSSr-based hybrid structures) embedded into D24WHFSS. The ${}^{RankTPFs}TU_{1:28}$ are fed to four zones (Z^1 to Z^4) decorated by the bi-IWM-oriented six-type L4SH structures (6L4SH) (See Fig. 2, e.g., Zone¹ including L4SH¹ to L4SH⁶) called 24WHFSS. By setting DKFPs (See Fig. 2, sprinkler symbols) in the form of KFP^1 and KFP^2 (20), the 24WHFSS is performed in the two-fold repetition named D24WHFSS (1D24WHFSS and 2D24WHFSS). For example, Table 6 shows the results of entering ${}^{RankTPFs}TU_2$ into 1D24WHFSS . By exerting the first zone led by $IWSS^{SVM}$, the OFs per L4SH (L4SH^{1:6}) are selected (See Table 6, 3rd row), and the ${}^1D24WHFSS\text{-}2Z^1RTPFs$ set by intersecting $Z^1[OFs\text{-}L4SH^h]_{h=1:6}$ is obtained (See Table 6, 5th row). In Z^1 of 1D24WHFSS , in the case of L4SH⁵ structure, first, the ${}^{RankTPFs}TU_2$ are entered into the first stage of L4SH⁵ (See Fig. 2, Z^1 -L4SH⁵, first circle blue-face tree), namely $SU\text{IWSSr}^{RBF\text{SVM}}$. The OFs of the first stage of L4SH⁵ (${}^1L4SH^5$) is $OFs\text{-}{}^1L4SH^5 = \{TPF_1, TPF_3, TPF_4\}$ with 87.80% classification accuracy (accuracy (Acc) metric in Table 9). Next, the ranked $OFs\text{-}{}^1L4SH^5$ is entered into $SU\text{IWSSr}^{RBF\text{TWSVM}}$ (See Fig. 2, Z^1 -L4SH⁵, circle²: yellow-face tree), and $OFs\text{-}{}^2L4SH^5 = \{TPF_1, TPF_4\}$: 82.92% is obtained. In the third stage, the sorted ${}^2L4SH^5$ are fed

to $SU\text{IWSSr}^{RBF\text{TWSVM}}$, and ${}^3L4SH^5 = \{TPF_1, TPF_4\}$: 82.92% is obtained (See Fig. 2, Z^1 -L4SH⁵, circle³ with orange-face tree). In the last round, ${}^3L4SH^5$ are entered into $SU\text{IWSSr}^{RBF\text{SVM}}$ (See Fig. 2, Z^1 -L4SH⁵, circle⁴ with green-face tree), and the result is recorded as ${}^4L4SH^5 = \{TPF_1, TPF_4\}$: 90.24%. The ${}^4L4SH^5$ is the TU_2 -specific $OFs\text{-}L4SH^5$ of Z^1 (See Table 6, row 3, column 5) based on the 1D24WHFSS setting. Such a mechanism is performed in the rest chained hybrid scenarios (L4SH^{1:4} and L4SH⁶) for extracting $OFs\text{-}L4SH^{1:4}$ and $OFs\text{-}L4SH^6$ (See Table 6, row 3, column 1 to column 4 and column 6). In Z^2 of 1D24WHFSS led by $IWSS^{TWSVM}$, the 6L4SH of Z^2 is exerted, and Z^2 -specific OFs are obtained (See Table 6, 8th row) and recorded in the ${}^1D24WHFSS\text{-}2Z^2RTPFs$ array by the intersection of $Z^2[OFs\text{-}L4SH^h]_{h=1:6}$ (See Table 6, 10th row). The optimal features obtained by Z^3 ($IWSSr^{SVM}$ as the starter of 6L4SH) and Z^4 ($IWSSr^{TWSVM}$ is the leader in 6L4SH) of 1D24WHFSS are gathered in ${}^1D24WHFSS\text{-}2Z^3RTPFs$ and ${}^1D24WHFSS\text{-}2Z^4RTPFs$, respectively. Next, by union the ${}^1D24WHFSS\text{-}2Z^{1:4}RTPFs$ ($\cup [{}^1D24WHFSS\text{-}2Z^iRTPFs]_{i=1:4}$), the ${}^1D24WHFSS\text{-}2Z^{1:4}URTPFs$ set containing $\{TPF_1, TPF_4, TPF_5\}$ is obtained (See Table 6, last row or See Table 7, 3th row, column 1). According to what was mentioned on obtaining ${}^1D24WHFSS\text{-}2Z^{1:4}URTPFs$, by setting the KFP^2 on 24WHFSS, the execution of 2D24WHFSS is lead to obtaining the ${}^2D24WHFSS\text{-}2Z^{1:4}URTPFs$ (See Table 7, 3th row, column 2). As can be seen in Table 7, the ${}^1D24WHFSS\text{-}xZ^{1:4}URTPFs$ and ${}^2D24WHFSS\text{-}xZ^{1:4}URTPFs$ sets related to each ${}^{RankTPFs}TU_x$ (${}^{RankTPFs}TU_{1:28}$) are listed. The xMRTPFs -specific results (1MRTPFs to ${}^{28}MRTPFs$) stemmed from union result of the ${}^1D24WHFSS\text{-}xZ^{1:4}URTPFs$ and ${}^2D24WHFSS\text{-}xZ^{1:4}URTPFs$ ($\cup [{}^1D24WHFSS\text{-}$

TABLE 7. Selecting ${}^1\text{D24WHFSS-}^x\text{Z}^{1:4}\text{URTPFs}$ of RankTPFsTU_x based on D24WHFSS.

Input	${}^1\text{D24WHFSS-}^x\text{Z}^{1:4}\text{URTPFs}$	${}^2\text{D24WHFSS-}^x\text{Z}^{1:4}\text{URTPFs}$
RankTPFsTU_1	$\{{}^1\text{TPF}_6\}$	$\{{}^1\text{TPF}_1, {}^1\text{TPF}_5, {}^1\text{TPF}_6\}$
RankTPFsTU_2	$\{{}^2\text{TPF}_1, {}^2\text{TPF}_4, {}^2\text{TPF}_5\}$	$\{{}^2\text{TPF}_1\}$
RankTPFsTU_3	$\{{}^3\text{TPF}_1, {}^3\text{TPF}_4, {}^3\text{TPF}_6\}$	$\{{}^3\text{TPF}_5\}$
RankTPFsTU_4	$\{{}^4\text{TPF}_1, {}^4\text{TPF}_4\}$	$\{{}^4\text{TPF}_5\}$
RankTPFsTU_5	$\{{}^5\text{TPF}_1, {}^5\text{TPF}_5, {}^5\text{TPF}_6\}$	$\{{}^5\text{TPF}_1, {}^5\text{TPF}_2, {}^5\text{TPF}_6\}$
RankTPFsTU_6	$\{{}^6\text{TPF}_4\}$	$\{{}^6\text{TPF}_3, {}^6\text{TPF}_6\}$
RankTPFsTU_7	$\{{}^7\text{TPF}_1, {}^7\text{TPF}_5\}$	$\{{}^7\text{TPF}_1\}$
RankTPFsTU_8	$\{{}^8\text{TPF}_2, {}^8\text{TPF}_5, {}^8\text{TPF}_6\}$	$\{{}^8\text{TPF}_2\}$
RankTPFsTU_9	$\{{}^9\text{TPF}_1, {}^9\text{TPF}_2\}$	$\{{}^9\text{TPF}_1, {}^9\text{TPF}_5\}$
RankTPFsTU_{10}	$\{{}^{10}\text{TPF}_1\}$	$\{{}^{10}\text{TPF}_1\}$
RankTPFsTU_{11}	$\{{}^{11}\text{TPF}_2, {}^{11}\text{TPF}_3, {}^{11}\text{TPF}_4, {}^{11}\text{TPF}_5\}$	$\{{}^{11}\text{TPF}_1, {}^{11}\text{TPF}_3, {}^{11}\text{TPF}_5\}$
RankTPFsTU_{12}	$\{{}^{12}\text{TPF}_1, {}^{12}\text{TPF}_2\}$	$\{{}^{12}\text{TPF}_1, {}^{12}\text{TPF}_2, {}^{12}\text{TPF}_3\}$
RankTPFsTU_{13}	$\{{}^{13}\text{TPF}_1, {}^{13}\text{TPF}_4, {}^{13}\text{TPF}_5\}$	$\{{}^{13}\text{TPF}_{1:3}\}$
RankTPFsTU_{14}	$\{{}^{14}\text{TPF}_3, {}^{14}\text{TPF}_4, {}^{14}\text{TPF}_6\}$	$\{{}^{14}\text{TPF}_2, {}^{14}\text{TPF}_4\}$
RankTPFsTU_{15}	$\{{}^{15}\text{TPF}_1, {}^{15}\text{TPF}_2\}$	$\{{}^{15}\text{TPF}_1, {}^{15}\text{TPF}_5\}$
RankTPFsTU_{16}	$\{{}^{16}\text{TPF}_1, {}^{16}\text{TPF}_2\}$	$\{{}^{16}\text{TPF}_2, {}^{16}\text{TPF}_4\}$
RankTPFsTU_{17}	$\{{}^{17}\text{TPF}_4\}$	$\{{}^{17}\text{TPF}_4, {}^{17}\text{TPF}_5\}$
RankTPFsTU_{18}	$\{{}^{18}\text{TPF}_2, {}^{18}\text{TPF}_6\}$	$\{{}^{18}\text{TPF}_5, {}^{18}\text{TPF}_6\}$
RankTPFsTU_{19}	$\{{}^{19}\text{TPF}_4, {}^{19}\text{TPF}_5\}$	$\{{}^{19}\text{TPF}_4, {}^{19}\text{TPF}_5\}$
RankTPFsTU_{20}	$\{{}^{20}\text{TPF}_5, {}^{20}\text{TPF}_6\}$	$\{{}^{20}\text{TPF}_5\}$
RankTPFsTU_{21}	$\{{}^{21}\text{TPF}_1, {}^{21}\text{TPF}_6\}$	$\{{}^{21}\text{TPF}_1\}$
RankTPFsTU_{22}	$\{{}^{22}\text{TPF}_1, {}^{22}\text{TPF}_4, {}^{22}\text{TPF}_5\}$	$\{{}^{22}\text{TPF}_2, {}^{22}\text{TPF}_5\}$
RankTPFsTU_{23}	$\{{}^{23}\text{TPF}_3, {}^{23}\text{TPF}_6\}$	$\{{}^{23}\text{TPF}_2, {}^{23}\text{TPF}_6\}$
RankTPFsTU_{24}	$\{{}^{24}\text{TPF}_2\}$	$\{{}^{24}\text{TPF}_3, {}^{24}\text{TPF}_4\}$
RankTPFsTU_{25}	$\{{}^{25}\text{TPF}_3, {}^{25}\text{TPF}_4\}$	$\{{}^{25}\text{TPF}_5, {}^{25}\text{TPF}_6\}$
RankTPFsTU_{26}	$\{{}^{26}\text{TPF}_2, {}^{26}\text{TPF}_3, {}^{26}\text{TPF}_6\}$	$\{{}^{26}\text{TPF}_3, {}^{26}\text{TPF}_5\}$
RankTPFsTU_{27}	$\{{}^{27}\text{TPF}_2\}$	$\{{}^{27}\text{TPF}_1, {}^{27}\text{TPF}_2, {}^{27}\text{TPF}_4\}$
RankTPFsTU_{28}	$\{{}^{28}\text{TPF}_1, {}^{28}\text{TPF}_2, {}^{28}\text{TPF}_4\}$	$\{{}^{28}\text{TPF}_3\}$

Symbol: TPF: transient point feature; TU_x : x^{th} transient univariate; ${}^x\text{TPF}_j$: j^{th} TPFs of TU_x ; RankTPFsTU_x : Sorted TPFs of TU_x based on SU; ${}^i\text{D24WHFSS-}^x\text{Z}^{1:4}\text{URTPFs}$: Selecting ${}^x\text{Z}^{1:4}\text{URTPFs}$ by performing ${}^i\text{D24WHFSS}$ based on KFP; ${}^x\text{Z}^{1:4}\text{URTPFs}$: contain union result of Zone¹-specific RTPFs to Zone^x-specific RTPFs per ${}^i\text{D24WHFSS}$.

TABLE 8. The obtained $\text{UMRTPFs}^{1:28}$ set based on union ${}^1\text{MRTPFs}$ to ${}^{28}\text{MRTPFs}$.

Input	${}^x\text{MRTPFs}$	Input	${}^x\text{MRTPFs}$
$\cup [{}^i\text{D24WHFSS-}^x\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^1\text{TPF}_1, {}^1\text{TPF}_5, {}^1\text{TPF}_6\}$	$\cup [{}^i\text{D24WHFSS-}^{15}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{15}\text{TPF}_1, {}^{15}\text{TPF}_2, {}^{15}\text{TPF}_3\}$
$\cup [{}^i\text{D24WHFSS-}^2\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^2\text{TPF}_1, {}^2\text{TPF}_4, {}^2\text{TPF}_5\}$	$\cup [{}^i\text{D24WHFSS-}^{16}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{16}\text{TPF}_1, {}^{16}\text{TPF}_2, {}^{16}\text{TPF}_4\}$
$\cup [{}^i\text{D24WHFSS-}^3\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^3\text{TPF}_1, {}^3\text{TPF}_{4,6}\}$	$\cup [{}^i\text{D24WHFSS-}^{17}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{17}\text{TPF}_4, {}^{17}\text{TPF}_5\}$
$\cup [{}^i\text{D24WHFSS-}^4\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^4\text{TPF}_1, {}^4\text{TPF}_4, {}^4\text{TPF}_5\}$	$\cup [{}^i\text{D24WHFSS-}^{18}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{18}\text{TPF}_2, {}^{18}\text{TPF}_5, {}^{18}\text{TPF}_6\}$
$\cup [{}^i\text{D24WHFSS-}^5\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^5\text{TPF}_1, {}^5\text{TPF}_2, {}^5\text{TPF}_5, {}^5\text{TPF}_6\}$	$\cup [{}^i\text{D24WHFSS-}^{19}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{19}\text{TPF}_4, {}^{19}\text{TPF}_5\}$
$\cup [{}^i\text{D24WHFSS-}^6\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^6\text{TPF}_3, {}^6\text{TPF}_4, {}^6\text{TPF}_6\}$	$\cup [{}^i\text{D24WHFSS-}^{20}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{20}\text{TPF}_3, {}^{20}\text{TPF}_5\}$
$\cup [{}^i\text{D24WHFSS-}^7\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^7\text{TPF}_1, {}^7\text{TPF}_5\}$	$\cup [{}^i\text{D24WHFSS-}^{21}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{21}\text{TPF}_1, {}^{21}\text{TPF}_6\}$
$\cup [{}^i\text{D24WHFSS-}^8\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^8\text{TPF}_2, {}^8\text{TPF}_5, {}^8\text{TPF}_6\}$	$\cup [{}^i\text{D24WHFSS-}^{22}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{22}\text{TPF}_1, {}^{22}\text{TPF}_2, {}^{22}\text{TPF}_4, {}^{22}\text{TPF}_5\}$
$\cup [{}^i\text{D24WHFSS-}^9\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^9\text{TPF}_1, {}^9\text{TPF}_2, {}^9\text{TPF}_5\}$	$\cup [{}^i\text{D24WHFSS-}^{23}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{23}\text{TPF}_2, {}^{23}\text{TPF}_3, {}^{23}\text{TPF}_6\}$
$\cup [{}^i\text{D24WHFSS-}^{10}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{10}\text{TPF}_1\}$	$\cup [{}^i\text{D24WHFSS-}^{24}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{24}\text{TPF}_{2:4}\}$
$\cup [{}^i\text{D24WHFSS-}^{11}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{11}\text{TPF}_{1:5}\}$	$\cup [{}^i\text{D24WHFSS-}^{25}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{25}\text{TPF}_{3:6}\}$
$\cup [{}^i\text{D24WHFSS-}^{12}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{12}\text{TPF}_{1:3}\}$	$\cup [{}^i\text{D24WHFSS-}^{26}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{26}\text{TPF}_2, {}^{26}\text{TPF}_3, {}^{26}\text{TPF}_5, {}^{26}\text{TPF}_6\}$
$\cup [{}^i\text{D24WHFSS-}^{13}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{13}\text{TPF}_{1:5}\}$	$\cup [{}^i\text{D24WHFSS-}^{27}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{27}\text{TPF}_1, {}^{27}\text{TPF}_2, {}^{27}\text{TPF}_4\}$
$\cup [{}^i\text{D24WHFSS-}^{14}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{14}\text{TPF}_{2,4}, {}^{14}\text{TPF}_6\}$	$\cup [{}^i\text{D24WHFSS-}^{28}\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$	$\{{}^{28}\text{TPF}_{1:4}\}$

UMRTPFs^{1:28}: Union of ¹MRTPFs to ²⁸MRTPFs

$$\{{}^1\text{TPF}_{1,5,6}, {}^2\text{TPF}_{1,4,5}, {}^3\text{TPF}_{1,4,6}, {}^4\text{TPF}_{1,4,5}, {}^5\text{TPF}_{1,2,5,6}, {}^6\text{TPF}_{3,4,6}, {}^7\text{TPF}_{1,5}, {}^8\text{TPF}_{2,5,6}, {}^9\text{TPF}_{1,2,5}, {}^{10}\text{TPF}_1, {}^{11}\text{TPF}_{1:5}, {}^{12}\text{TPF}_{1:3}, {}^{13}\text{TPF}_{1:5}, {}^{14}\text{TPF}_{2,4,6}, {}^{15}\text{TPF}_{1,2,5}, {}^{16}\text{TPF}_{1,2,4}, {}^{17}\text{TPF}_{4,5}, {}^{18}\text{TPF}_{2,5,6}, {}^{19}\text{TPF}_{4,5}, {}^{20}\text{TPF}_{3,5}, {}^{21}\text{TPF}_{1,6}, {}^{22}\text{TPF}_{1,2,4,5}, {}^{23}\text{TPF}_{2,3,6}, {}^{24}\text{TPF}_{2,4}, {}^{25}\text{TPF}_{3,6}, {}^{26}\text{TPF}_{2,3,5,6}, {}^{27}\text{TPF}_{1,2,4}, {}^{28}\text{TPF}_{1:4}\}$$

Symbol: TPF: transient point feature; TU_x : x^{th} transient univariate; ${}^x\text{TPF}_j$: j^{th} TPFs of TU_x ; $\cup [{}^i\text{D24WHFSS-}^x\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$: Selecting ${}^x\text{MRTPFs}$ based on union of ${}^i\text{D24WHFSS}$ -specific ${}^x\text{Z}^{1:4}\text{URTPFs}$ and ${}^i\text{D24WHFSS}$ -specific ${}^x\text{Z}^{1:4}\text{URTPFs}$.

${}^x\text{Z}^{1:4}\text{URTPFs}]_{i=1:2}$ is shown in Table 8. Finally, $\text{UMRTPFs}^{1:28}$ is obtained by the union of ${}^1\text{MRTPFs}$ to ${}^{28}\text{MRTPFs}$ (See Table 8, last row).

In the case of exerted wrapper-based predictive models (SVM and TWSVM) in D24WHFSS, the following points are important. The accuracy (Acc) metric (21) measured the per-

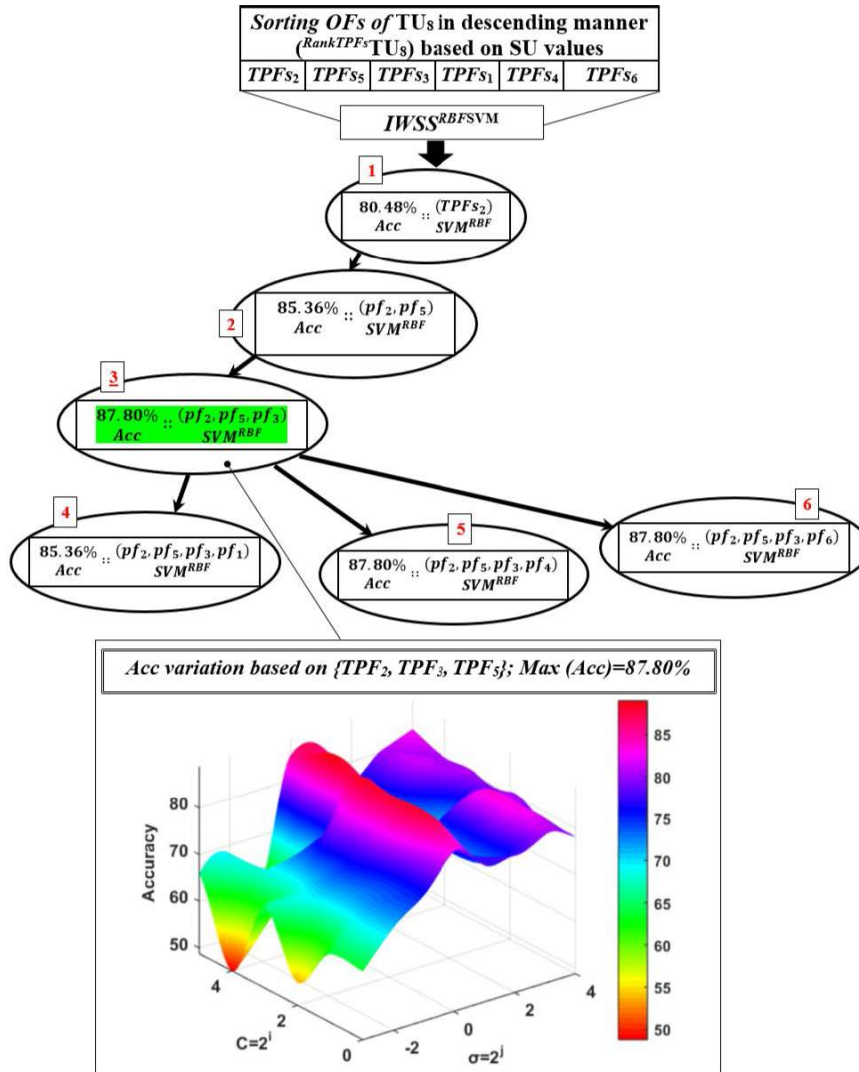


FIGURE 9. Structure of $IWSS^{RBF SVM}$ tree related to Z^1 -specific first stage of $L4SH^{1:6}$ of 1D24WHFSS in selecting OFs of TU_8 regarding Acc variations in optimal node (node 3).

TABLE 9. The evaluation metrics.

Metrics	Formula
Accuracy	$Acc = (TP + TN) / (TP + TN + FP + FN)$
Sensitivity	$TPR = TP / (TP + FN)$
Specificity	$TNR = TN / (TN + FP)$

formance of $SU IWSS^{k SVM}$, $SU IWSS_r^{k SVM}$, $SU IWSS^k TWSVM$, and $SU IWSS_r^k TWSVM$ learning models. On the other hand, the fine-tuning of learning parameters (C in SVM and TWSVM, σ in RBF and DTW, and p in Poly) in each increment of bi-IWM is considered in train-test procedures. The range of learning parameters (RoLPs) related to SVM (SVM^{RoLPs}) and TWSVM ($TWSVM^{RoLPs}$) is defined in (22) and (23), respectively. In each iteration, the maximum amount of the Acc which is obtained based on the best value of learning parameters is recorded. For example, Fig. 9 shows

the $IWSS^{RBF SVM}$ tree related to Z^1 -specific first stage of $L4SH^{1:6}$ of 1D24WHFSS applied on TU_8 , which Acc variations of optimal node (node 3: green-face) is depicted in 3-D plot.

$$QKFPs = \left\{ \begin{array}{l} KFP^1 : (RBF, RBF) | SVM^{RBF}, TWSVM^{RBF} \\ KFP^2 : (DTW, Poly) | SVM^{DTW}, TWSVM^{Poly} \end{array} \right\} \quad (20)$$

$$Accuracy(Acc) = (TP + TN) / (TP + TN + FP + FN) \quad (21)$$

$$\left\{ \begin{array}{l} P : \text{stable sample}; T : \text{predicted correctly} \\ N : \text{unstable sample}; F : \text{predicted incorrectly} \end{array} \right.$$

$$SVM^{RoLPs} \left\{ \begin{array}{l} C = 2^i \mid i = 0, 1, \dots, 5 \\ \sigma = 2^j \mid j = -3, -2, \dots, 4 (RBF, DTW) \end{array} \right\} \quad (22)$$

$$TWSVM^{RoLPs} \left\{ \begin{array}{l} C = 2^i \mid i = 0, 1, \dots, 5 \\ \sigma = 2^j \mid j = -3, -2, \dots, 4 (RBF) \\ p = 2^j \mid j = 2, 3, 4, 5 (Poly) \end{array} \right\} \quad (23)$$

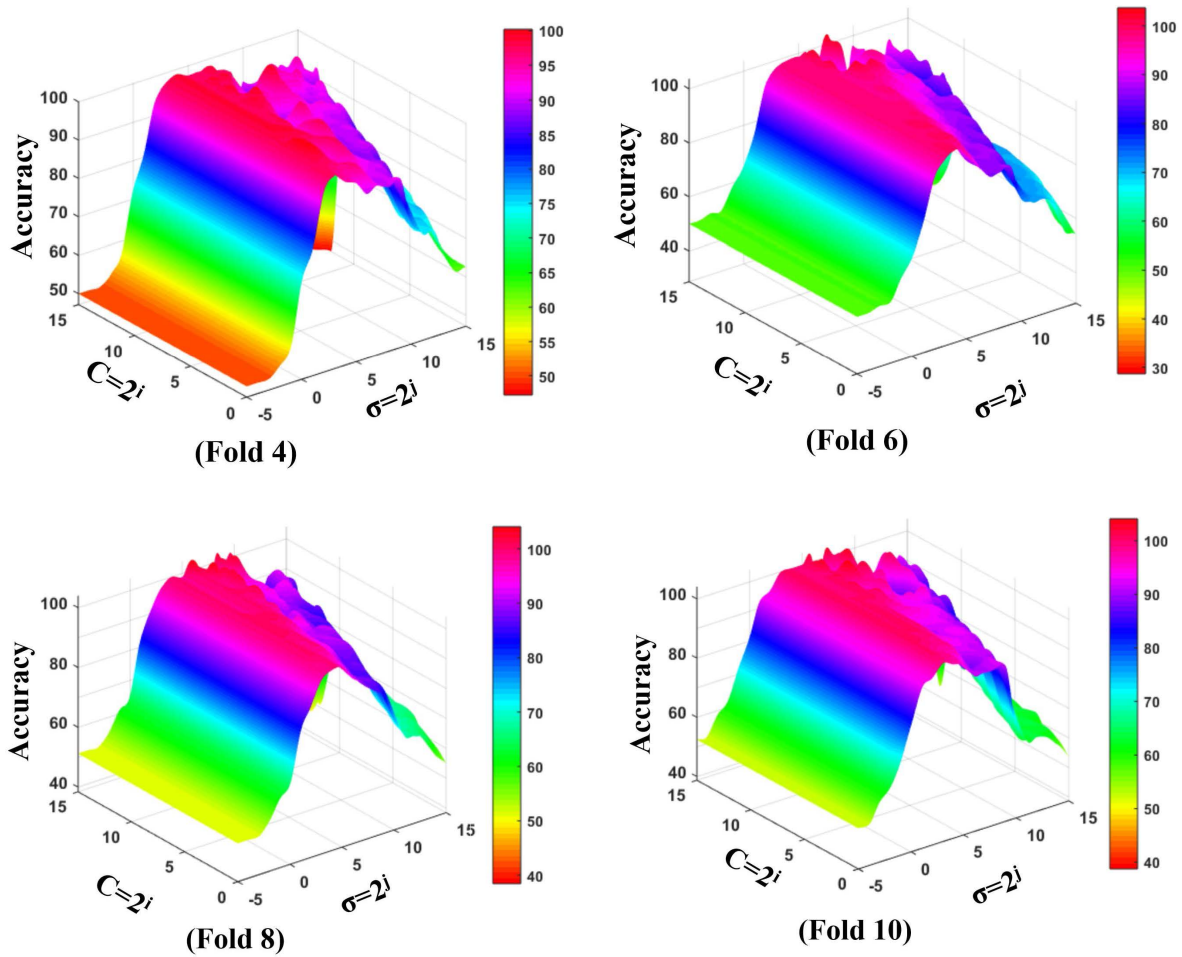


FIGURE 10. Acc variations based on learning parameters in some folds (fold^{4,6,8,10}) for TSP based on UMRTPFs^{1:28}.

TABLE 10. Results of TSP based on UMRTPFs^{1:28} set.

Classifier	Test case	10-fold cross validation			
		Max(Acc.) per fold based on fine-tuning on C and σ Accuracy [TPR / TNR]			
SVM (RBF)	NETS-NYPS	fold 1	fold 2	fold 3	fold 4
		98.75	96.25	98.75	100
		[97.5 / 100]	[97.5 / 95]	[97.5 / 100]	[100 / 100]
		fold 5	fold 6	fold 7	fold 8
		100	100	100	100
		[100 / 100]	[100 / 100]	[100 / 100]	[100 / 100]
		fold 9	fold 10		
		98.75	100		
		[97.5 / 100]	[100 / 100]		
		Mean (measure) of folds: Accuracy [TPR / TNR]			
99.25 [99 / 99.5]					

TABLE 11. TPT based on coupling UMRTPFs^{1:28} and SVM^{RBF}.

Observation Window Time (OWT) in cycle / second	TPT (OWT + prediction time)
6 / 0.1002	100.2 ms + 2.407 ms = 102.607 ms

C. TSP BASED ON UMRTPFs^{1:28} SET

The union of TU_{1:28}-specific MRTPFs (^{1:28}MRTPFs) called UMRTPFs^{1:28} (See Table 8, last row) is used for TSP in this section. The 10-fold cross-validation-based scenario

is considered for measuring the performance of the UMRTPFs^{1:28} on TSP. The SVM^{RBF}-based learning model is performed per fold-specific train-test procedure. Furthermore, the fine-tuning of the SVM^{RBF} parameters, namely C and σ , is considered regarding the $\{C = 2^i | i = 0, 1, \dots, 15\}$ and $\{\sigma = 2^j | j = -5, -4, \dots, 15\}$ to report the best values of evaluation metrics (See Table 9) per fold. According to the above-mentioned experimental design, the performance of SVM^{RBF} based on UMRTPFs^{1:28} is shown in Table 10. Set different values on learning parameters of

TABLE 12. Results of TSP via coupling SVM^{RBF} and selected OFs by 3MWHFSSs and 3MCWHFSSs

Classifier-FSS	Test case	10-fold cross validation			
		Max(Acc.) per fold based on fine-tuning on C and σ Accuracy [TPR / TNR]			
SVM ^{RBF} -mRMR	NETS-NYPS	fold 1	fold 2	fold 3	fold 4
		93.75	93.75	90	90
		[97.5 / 90]	[97.5 / 90]	[95 / 85]	[82.5 / 97.5]
		fold 5	fold 6	fold 7	fold 8
		95	95	95	91.25
		[97.5 / 92.5]	[95 / 95]	[100 / 90]	[87.5 / 95]
		fold 9	fold 10		
		88.75	93.75		
		[95 / 82.5]	[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	95	96.25	97.5
		[97.5 / 100]	[95 / 95]	[92.5 / 100]	[95 / 100]
		fold 5	fold 6	fold 7	fold 8
		96.25	97.5	97.5	97.5
		[92.5 / 100]	[97.5 / 97.5]	[95 / 100]	[97.5 / 97.5]
		fold 9	fold 10		
		97.5	100		
		[95 / 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	95	96.25	97.5
		[97.5 / 100]	[95 / 95]	[92.5 / 100]	[95 / 100]
		fold 5	fold 6	fold 7	fold 8
		96.25	97.5	97.5	97.5
		[92.5 / 100]	[97.5 / 97.5]	[95 / 100]	[97.5 / 97.5]
		fold 9	fold 10		
		97.5	100		
		[95 / 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	97.5	96.25	95
		[100 / 100]	[100 / 95]	[92.5 / 100]	[92.5 / 97.5]
		fold 5	fold 6	fold 7	fold 8
		100	100	97.5	98.75
		[100 / 100]	[100 / 100]	[97.5 / 97.5]	[97.5 / 100]
		fold 9	fold 10		
		97.5	100		
		[97.5 / 97.5]	[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	96.25	97.5	98.75
		[97.5 / 100]	[100 / 92.5]	[95 / 100]	[100 / 97.5]
		fold 5	fold 6	fold 7	fold 8
		100	100	98.75	100
		[100 / 100]	[100 / 100]	[97.5 / 100]	[100 / 100]
		fold 9	fold 10		
		97.5	100		
		[97.5 / 97.5]	[100 / 100]		
		Mean(measure) of folds: Accuracy [TPR / TNR]			
98.75 [98.75 / 98.75]					
		fold 1	fold 2	fold 3	fold 4
		98.75	95	97.5	98.75
		[97.5 / 100]	[95 / 95]	[95 / 100]	[100 / 97.5]

TABLE 12. (Continued.) Results of TSP via coupling SVM^{RBF} and selected OFs by 3MWHFSSs and 3MCWHFSSs.

Classifier-FSS	Test case	10-fold cross validation			
		Max(Acc.) per fold based on fine-tuning on C and σ Accuracy [TPR / TNR]			
		fold 5	fold 6	fold 7	fold 8
SVM ^{RBF} -CPQHFSS	NETS-NYPS	100	100	100	100
		[100 / 100]	[100 / 100]	[100 / 100]	[100 / 100]
		fold 9		fold 10	
		98.75	100		
		[97.5 / 100]	[100 / 100]		
		Mean(measure) of folds: Accuracy [TPR / TNR]			
		98.87 [98.5 / 99.25]			

TABLE 13. TPT based on 3MWHFSSs and 3MCWHFSSs.

SVM ^{RBF} - 3MWHFSSs / 3MCWHFSSs	OWT in cycle / second	TPT (OWT + 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
SVM ^{RBF} -CPQHFSS	9 / 0.1503	150.3 ms+2.225 ms=152.525 ms

TABLE 14. Amount of memory usage by Matlab-based TSPP.

Memory status	Max Possible Array Bytes	Mem Available All Arrays (MAAAs)	Mem Used MATLAB
Before starting TSPP (^{BS} TSPP)	3.4058e+09	3.4058e+09	1.2423e+09
After ending TSPP (^{AE} TSPP)	3.2955e+09	3.2955e+09	1.2503e+09
Memory used in Megabytes			
^{AE} TSPP.MAAAs - ^{BS} TSPP.MAAAs = - 105.25 Megabytes			

(-): indicates that the free memory is (about 105 megabytes) lower now than it was before started TSPP.

SVM^{RBF} cause obtaining various Acc per fold. As can be seen in Table 10, the maximum value of Acc is considered as the result of Acc-based performance evaluation per fold. For more clarity, the Acc variations in some folds (fold⁴, fold⁶, fold⁸, and fold¹⁰) are illustrated in Fig. 10. Also, the maximum Acc-specific TPR and TNR per fold are listed in Table 10. Finally, the mean value of obtained results in all folds per metric is calculated (See Table 10, last row). The Acc 99.25 %, TPR 99 %, and TNR 99.5 % indicates the high TPA on TSP via UMRTPFs^{1:28}. Another main factor in proving the efficiency of the UMRTPFs^{1:28}-oriented learning model is the TPT index (including observation window time (OWT) and prediction time). For TPT calculation, first, we focus on the TPFs of UMRTPFs^{1:28} (See Table 8, last row) to specify OWT. The most extended observed cycle in UMRTPFs^{1:28} is related to TPFs₆, which picked up as the optimal cycle of TU₁, TU₃, TU₅, TU₆, TU₈, TU₁₄, TU₁₈, TU₂₁, TU₂₃, TU₂₅, and TU₂₆. In this manner, the OWT is six cycles (100.2 milliseconds (ms)). On the other hand, the prediction time based on UMRTPFs^{1:28}-

SVM^{RBF} is 2.407 ms. Consequently, the TPT is 102.607 ms (See Table 11), reflecting the low TPT to exert control actions.

D. COMPARISON OF EXPERIMENTAL METHODS: D24WHFSS VS. 3MWHFSSs AND 3MCWHFSSs

For a deep assessment of the efficiency of the proposed FSS in selecting OFs, D24WHFSS is compared with 3MWHFSSs. The 3MWHFSSs includes mRMR [13], ReliefF [15] and FCBF [16]. Also, D24WHFSS is compared with 3MCWHFSSs including BMHFSS [19], CPQHFSS [22], and PITHS [23]. The 28VTTFs are fed to the 3MWHFSSs and 3MCWHFSSs, the 3MWHFSSs-based OFs and 3MCWHFSSs-based OFs are selected. Then, the 3MWHFSSs^{OFs} and 3MCWHFSSs^{OFs} are entered into the SVM^{RBF} based on similar train-test conditions defined for D24WHFSS (See Section IV. C).

As can be seen in Table 12, D24WHFSS-based UMRTPFs^{1:28} have better performance in TSP than 3MWHFSSs^{OFs} and 3MCWHFSSs^{OFs}. According to Table 12, the obtained results manifested the D24WHFSS by selecting 88-cycles of 28VTTFs (See Table 8, last row), which has better performance (Acc, TPR, and TNR) than mRMR^{OFs} (9 cycles-4VTTFs), FCBF^{OFs}, ReliefF^{OFs}, and BMHFSS^{OFs} (9 cycles-3VTTFs) [19], CPQHFSS^{OFs} (48 cycles-28VTTFs [22]), and PITHS^{OFs} (24 cycles-18VTTFs [23]). Based on the TPT index, the obtained results (See Table 13) show that the coupling D24WHFSS^{OFs} and SVM^{RBF} has a higher TPT (102.607 ms) than SVM^{RBF}-3MWHFSSs^{OFs} (SVM^{RBF}-mRMR^{OFs}: 68.793 ms, SVM^{RBF}-FCBF^{OFs}: 68.930 ms, SVM^{RBF}-ReliefF^{OFs}: 68.910 ms) and SVM^{RBF}-BMHFSS^{OFs} with 52.948 ms. Also, SVM^{RBF}-D24WHFSS^{OFs} has lower TPT than SVM^{RBF}-PITHS^{OFs} with 152.591 ms and SVM^{RBF}-CPQHFSS^{OFs} with 152.525 ms. The TPT value of SVM^{RBF}-D24WHFSS^{OFs} provides the proper opportunity for the grid operator to conduct corrective actions. For more TPT-based info, refer to Table 11 and Table 13. As the final report, checking the amount of memory usage by the transient status predictive program (TSPP) (MATLAB-based TSPP) is presented in the form of the user-focused memory structure in Table 14.

V. CONCLUSION AND FUTURE WORK

Thinking critically about the low-performance of the proposed mono-way hybrid FSS on TSA studies motivated us to design a novel feature selection algorithm called dyadic 24-way hybrid FSS (D24WHFSS) in this paper. The D24WHFSS is driven by the beating heart of linked four-level hybrid models (LFLMs) that the different permutations of levels cause execution LFLMs in 24-way (called 24WHFSS). The 24WHFSS is mounted on the bi-incremental wrapper mechanism (bi-IWM), namely IWSS and IWSSr. The filter and wrapper phases of bi-IWM are accompanied by SU and DSVCs, respectively. k SVM and k TWSVM are supervised machine learning algorithms regarded as DSVCs plugged into the bi-IWM. For precise mining on nonlinear HDTS, DKFPs are situated into DSVCs. Hence, KFPs-based 24WHFSS exerting is repeated in varied two times (dyadic 24WHFSS). After conducting D24WHFSS on m VTTFs, survived MRTPFs are entered into the cross-validation procedure to measure the efficacy of MRTPFs set in achieving low TPT and high TPA. Obtained results manifested that the MRTPFs have high performance (Acc 99.25 %, TPR 99 %, TNR 99.5 %, and TPT of 102.607 ms) for TSP. To address the effectiveness of the D24WHFSS against other feature selection algorithms, the performance of D24WHFSS compared with 3MWHFSSs and 3MCWHFSSs. The results show that selected MRTPFs by D24WHFSS have better performance than extracted optimal features by 3MWHFSSs and 3MCWHFSSs on TSP.

In future work, we intend to introduce a novel feature selection-feature extraction algorithm decorated by embedding a hybrid-based optimum-features selector layer in the convolutional deep network-based feature extraction. Such a framework can promise to pick up the most discriminative-relevant features on HDTS for high-performance TSA. Furthermore, the efficacy of the optimal transient features in achieving high-performance TSA under the N - k contingency analysis, load-generation level variations, and contaminated transient responses (missing and noisy transient data) is evaluated in future studies.

REFERENCES

- [1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
- [2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.
- [3] Y. Wang, X. Wang, Y. Wu, and Y. Guo, "Power system fault classification and prediction based on a three-layer data mining structure," *IEEE Access*, vol. 8, pp. 200897–200914, 2020.
- [4] W. N. Ismail, M. M. Hassan, H. A. Alsalamah, and G. Fortino, "CNN-based health model for regular health factors analysis in internet-of-medical things environment," *IEEE Access*, vol. 8, pp. 52541–52549, 2020.
- [5] N. O. Alshrein, A. F. Klaib, and A. Magableh, "Intelligent transportation and control systems using data mining and machine learning techniques: A comprehensive study," *IEEE Access*, vol. 7, pp. 49830–49857, 2019.
- [6] G. V. Trunk, "A problem of dimensionality: A simple example," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 3, pp. 306–307, Jul. 1979.
- [7] K. Koutroumbas and S. Theodoridis, *Pattern Recognition*, 1st ed. New York, NY, USA: Academic, 2008.
- [8] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 1st ed. New York, NY, USA: Springer, 2013.
- [9] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos, *Feature Selection for High-Dimensional Data*, 1st ed. Cham, Switzerland: Springer, 2016.
- [10] P. Kundur, "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.
- [11] C.-S.-G. Karavas, K. A. Plakas, K. F. Krommydas, A. S. Kurashvili, C. N. Dikaiakos, and G. P. Papaioannou, "A review of wide-area monitoring and damping control systems in Europe," in *Proc. IEEE Madrid PowerTech*, Jun. 2021, pp. 1–6.
- [12] 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.
- [13] 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, 2019.
- [14] 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.
- [15] 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)*, Aug. 2018, pp. 171–176.
- [16] J. Yan, C. Li, and Y. Liu, "Deep learning based total transfer capability calculation model," in *Proc. Int. Conf. Power Syst. Technol. (POWERCON)*, Nov. 2018, pp. 952–957.
- [17] 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.
- [18] 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.
- [19] S. A. B. 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, 2021.
- [20] Y. Li and Z. Yang, "Application of EOS-ELM with binary Jaya-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23092–23101, 2017.
- [21] X. Gu, Y. Li, and J. Jia, "Feature selection for transient stability assessment based on kernelized fuzzy rough sets and memetic algorithm," *Int. J. Elect. Power Energy Syst.*, vol. 64, pp. 664–670, Jan. 2015.
- [22] S. A. B. 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, 2022.
- [23] S. A. B. 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, 2021.
- [24] P. Bermejo, J. A. Gamez, and J. M. Puerta, "Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Mar. 2009, pp. 367–374.
- [25] D. Tomar and S. Agarwal, "Twin support vector machine," *Egyptian Inform. J.*, vol. 16, no. 1, pp. 55–69, Mar. 2015.
- [26] *MATLAB: The Language of Technical Computing*, 6th ed., Mathworks, Massachusetts, MA, USA, 2000.
- [27] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Incremental wrapper-based gene selection from microarray data for cancer classification," *Pattern Recognit.*, vol. 39, no. 12, pp. 2383–2392, Dec. 2006.
- [28] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington, MA, USA: Morgan Kaufmann, 2011.
- [29] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, Apr. 1995.

- [30] H. Shimodaira, K.-I. Noma, M. Nakai, and S. Sagayama, "Support vector machine with dynamic time-alignment kernel for speech recognition," in *Proc. 7th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, Sep. 2001, pp. 1–4.
- [31] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.
- [32] R. Khemchandani and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, Mar. 2007.
- [33] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Cham, Switzerland: Springer, 2006.
- [34] 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>
- [35] C. Canizares, T. Fernandes, E. Geraldi, 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.
- [36] A. B. Mosavi, A. Amiri, and H. Hosseini, "A learning framework for size and type independent transient stability prediction of power system using twin convolutional support vector machine," *IEEE Access*, vol. 6, pp. 69937–69947, 2018.
- [37] S. A. B. Mosavi, "Extracting discriminative features in reactive power variations by 1-persistence parallel fragmented hybrid feature selection scheme for transient stability prediction," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 4, pp. 500–513, Aug. 2021.
- [38] S. A. B. Mosavi and S. Y. Banihashem, "Defining geometric cross-relevance multivariate trajectory features for transient stability analysis based on elastic kernel support vector machine," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 5, pp. 369–385, Oct. 2021.



SEYED ALIREZA BASHIRI MOSAVI received the B.Sc. degree in computer engineering-software from the Azad University of Zanjan, Zanjan, Iran, in 2011, the M.Sc. degree in information technology engineering-electronic commerce from the University of Qom, Qom, Iran, in 2013, and the Ph.D. degree in computer engineering-artificial intelligence and robotics from the University of Zanjan, in 2019. His master's thesis on customer value analysis won the Tejarat Bank Award as the best thesis related to quality. His Ph.D. thesis on designing a novel learning framework for size and type independent transient stability prediction of the power systems. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Buein Zahra Technical University, Buein Zahra, Qazvin, Iran. He has published several papers and conferences in the field of customer relationship management (CRM), transient processes in power systems, and machine learning scope. His main research interests include CRM, data mining, pattern recognition, and transient analysis based on machine learning methods.

• • •