Applying Cross-Permutation-Based Quad-Hybrid Feature Selection Algorithm on Transient Univariate to Select Optimal Features for Transient Analysis

Seyed Alireza Bashiri Mosavi
Department of Electrical and Computer Engineering, Buein Zahra Technical University, Buein Zahra, Qazvin 3451866391, Iran
Corresponding author: Seyed Alireza Bashiri Mosavi (e-mail: abashirimosavi@bzte.ac.ir).

ABSTRACT
Neglect feature selection matter for high-dimensional transient data obtained from phasor measurement units (PMUs) negatively affects the inconsistent-linked indices, namely data labeling time (DLT) and data labeling accuracy (DLA) in the transient analysis (TA). A reasonable trade-off between DLT and DLA or a win-win solution (low DLT and high DLA) necessitates feature-based mining on transient multivariate excursions (TMEs) via designing the comprehensive feature selection scheme (FSS). Hence, to achieve high-performance TA, we offer the 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, namely incremental wrapper subset selection (IWSS) and IWSS with replacement (IWSSr). The IWSS\(^2\)FWBs and IWSSr\(^2\)FWBs contain filter-fixed and wrapper-varied approaches (F\(^W\)v) that first block-specific F\(^W\)v of IWSS\(^2\)FWBs and IWSSr\(^2\)FWBs includes relevancy ratio-support vector machine (RR-SVM) and second block-specific F\(^W\)v of IWSS\(^2\)FWBs and IWSSr\(^2\)FWBs accompanied by relevancy ratio-twin support vector machine (RR-TWSVM). Generally, RR\(^{IWSS}\)SVM and RR\(^{IWSS}\)TWSVM is in IWSS\(^2\)FWBs, and RR\(^{IWSSr}\)SVM and RR\(^{IWSSr}\)TWSVM is in IWSSr\(^2\)FWBs. Besides direct relations in two-F\(^W\)v per incremental wrapper mechanism, by plugging different kernels into the hyperplane-based wrapper, all possible cross-permutations of hybrid FSS are applied on transient data to extract the optimal transient features (OTFs). Finally, the evaluation of the effectiveness of the CPQHFSS-based OTFs in TA is conducted based on the cross-validation technique. The obtained results show that the proposed framework has a DLA of 98.87% and a DLT of 152.525 milliseconds for TA.

INDEX TERMS
Hybrid feature selection algorithm, Optimal transient features, Transient analysis.

ACRONYMS
PMUs Phasor Measurement Units.
TA Transient Analysis.
DLT Data Labeling Time.
DLA Data Labeling Accuracy.
TMEs Transient Multivariate Excursions.
FSS Feature Selection Scheme.
CPQHFSS Cross-Permutation-based Quad-Hybrid FSS.
FWBs Filter-Wrapper Blocks.
HBs Hybrid Blocks.
IWSS Incremental Wrapper Subset Selection.
IWSSr IWSS with Replacement.
RR Relevancy Ratio.
SU Symmetric Uncertainty.
SVM Support Vector Machine.
TWSVM Twin Support Vector Machine.
OTFs Optimal Transient Features.
IT Information Technology.
DM Data Mining.
ML Machine Learning.
HDD High-Dimensional Data.
TTP Training-Testing Procedure.
A&T Accuracy and Time.
OFs Optimal Features.
MDFs Most Discriminative Features.
TSA Transient Stability Assessment.
TSP Transient Stability Prediction.
I. INTRODUCTION

Nowadays, information technology (IT) by integrating different data-driven systems, plays the pivot role in collecting a large amount of data in different sensitive industries. The raw data obtained by the IT paradigm provide the necessary conditions for conducting data-oriented actions instead of experience-based operations in all tasks and responsibilities of system operators [1-5]. Such restructuring in the decision-making process will be possible through data mining (DM) technology which triangulated machine learning (ML), statistical learning (SL), and dataset to discover useful patterns for predicting different phenomena [6, 7]. Besides the importance of the type of ML and SL methods for achieving efficiency in learning procedures, the high-dimensional data (HDD) with sparse-dissimilar features is the most significant factor that negatively affects the training-testing procedures (TTP) of learning frameworks. In this regard, the concept of the curse of dimensionality is defined as a great challenge on the way of high-performance DM [8, 9]. Furthermore, the importance of inconsistent-linked indices like accuracy and time (A&T) to make A&T-critical prediction in real-world problems exacerbates the necessity of focusing on HDD concern. To fill two needs with one deed, DM engineers apply FSS on HDD for extracting the optimal features.
(OFs) set [10, 11]. The survived OFs based on the FSS will bring two points: first, the low processing time to predict unseen cases due to the mapping HDD to low-dimensional feature space, and second, high accuracy prediction induced by selecting most discriminative features (MDFs).

One of the HDD-oriented real-world problems is transient stability assessment (TSA) related to secure power supply [12-13]. The IT-based grid infrastructure equipped with phasor measurement units (PMUs) gathers high-dimensional transient features (HDTF) for transient analysis (TA) [14]. In HDTF, the presence of irrelevant and redundant features is problematic for TTP of predictive approaches, which cause low-accuracy transient stability prediction (TSP). By applying the FSS scheme on HDTF, optimal transient features (OTFs) are selected to achieve high data labeling accuracy (DLA) in TSP. Also, the severe-sudden essence of transient stability necessitates using the FSS for compacting the HDTF to decrease data labeling time (DLT) in TSP, including observed time and prediction time [15]. In terms of the low DLT, low dimensional optimal transient features (LDOTF) caused fast learning in TTP scenarios leads to low prediction time, and existing the most relevant features in LDOTF allow picking up small-optimal observations. Consequently, by applying the FSS scheme on HDTF, system operators will be able to take timely-accurate corrective control actions to provide secure-adequate exploitation of the power grid. Hence, to achieve a win-win trade-off (high DLA and low DLT), designing the comprehensive FSS has been widely considered by DM researchers for TA.

II. RELATED WORKS

Reviewing the FSS-based transient studies shows that optimal transient features are selected by filter and filter-wrapper (hybrid) methods. In term of filter-oriented FSS, in Reference [16, 17], mutual information theory applied on transient characteristics related to power and angle to select optimal features regarding two principles: selected transient features have maximum relevance to the target class and have minimum relevance to one another, which is called minimum-redundancy and maximum-relevance (mRMR) FSS. Reference [18] introduce the ReliefF algorithm for calculating relevancy of rotor faults features to predict the health state of induction motor. To calculate total transfer capability regarding transient stability limitations, designing the feature pre-screening scenario based on the fast correlation-based filter (FCBF) is considered in [19].

Based on FCBF, optimal features of active and reactive load power, phase angles of bus voltages, and the induced electromotive force of generators are selected to achieve training-testing efficiency. In Reference [20], for constructing the dynamic security assessment (DSA) model for predicting the transient stability margin, applying the FSS algorithm based on partial mutual information (PMI) and the Pearson correlation coefficient (PCC) is considered as the main step of the DSA model. In terms of hybrid FSSs, in Reference [21], filter-wrapper FSS includes feature weight ranking by Relief (filter as the preliminary step, and five-fold cross-validation SVM model (wrapper as the complementary step) are applied on trajectory cluster features for selecting optimal feature set. Based on the proposed hybrid FSS in [22], first, normalized mutual information (NMI) ranks the initial features in the form of strongly relevant feature subset (SRFS) and the weakly relevant feature subset (WRFS). Next, the obtained knowledge of the filter phase is fed to the wrapper phase equipped with an easy-implementing search algorithm called binary particle swarm optimization (BPSO) to improve the effectiveness of FSS results. Considering high-dimensional multivariate time series data obtained by transient simulations, Reference [23] designed hybrid FSS in bi-mode, including trajectory-based filter-wrapper method (TFWM) and point-based filter-wrapper method (PFWM). In TFWM, mutual information-entropy-based (MIE) calculations (filter) and fuzzy imperialist competitive algorithm (FICA)-IWSS-based trihedral kernel-SVM (wrapper) find the optimal transient series. Next, the PFWM, including MIE calculations (filter) and the Gaussian kernel-SVM (wrapper) utilized to find optimal point features per optimal time series.

Regardless of the precise mining on transient feature space by the abovementioned FSSs, which have led to the acceptable performance in TSA, designing the comprehensive hybrid framework to extract masked-relevant transient features is the greatest challenge to achieve timely-accurate TSA. Lack of cross-oriented learning mechanisms in the form of multifaceted hybrid FSS (MHFSS) causes some features with the discriminative character don't survive in the feature selection process. In this regard, focusing on the structure of the filter or hybrid FSSs in previous studies shows the fact that the mining of intrinsic characteristics of transient data for selecting optimal features is based on the unilateral strategy equipped with vertically learning. Such a mechanism may be applicable in selecting the optimal features to improve the TSA, but it will ignore optimal-blurred transient features. Furthermore, the characteristic of transient data is the main parameter in determining how to apply the proposed MHFSS to it. Having a glance at FSS-based studies shows that the FSSs applied on multivariate point or time series data in the whole-manner. Such a strategy tends to select the OTFs without regard to the possibility of optimal features sacrificing related to each univariate trajectory. Streaming k-variate time-series data obtained by the PMU-based synchronized measurement necessitate extracting univariate-specific OTFs in the form of univariate-oriented learning in both filter and wrapper phase.

The main contributions of this study to solve abovementioned challenges in FSS-based TSA are summarized as follows:
A new feature selection algorithm called cross-permutation quad-hybrid FSS (CPQHFSS) was considered to select OTFs to achieve high-performance TA. This scheme was designed via a multifaceted hybrid scenario accompanied by the entropy-based metric in the filter phase and hyperplane-based learning methods in the wrapper phase.

The CPQHFSS applied on high-dimensional TMEs based on partial-manner learning for extracting univariate-specific MDFs. The optimal features per univariates are survived by the conducting cross-permutation scenario of the proposed FSS. Such a mechanism guaranteed the optimal-blurred transient features extraction to achieve high DLA and low DLT.

The performance of CPQHFSS-specific OTFs in TA was compared with selected OTFs by other FSS based on the cross-validation technique.

The rest of the paper is organized as follows: The detailed descriptions of the CPQHFSS are remarked in Section 3. Experimental results of applying CPQHFSS on univariates of TMEs for TSA are presented in Section 4. Also, the comparison results between the proposed FSS and the other FSSs are interpreted in Section 4. Finally, the conclusion is depicted in Section 5.

III. CROSS-PERMUTATION-BASED QUAD-HYBRID FEATURE SELECTION SCHEME (CPQHFSS)

The overall framework of FSS-oriented TA based on CPQHFSS is shown in Fig. 1. After transient data gathering phase, we offer CPQHFSS including twin two-filter-wraper blocks (2FWBs) mounted on the two-mechanism of the incremental wrapper namely incremental wrapper subset selection (IWSS) and IWSS with replacement (IWSSr). The relevancy ratio-support vector machine (RR-SVM) is the filter-fixed wrapper-varied (FWv) embedded in the first block of IWSS and IWSSr. The second block-specific FWv of IWSS and IWSSr is designed by relevancy ratio-twin support vector machine (RR-TWSVM). Besides direct relations in four hybrid blocks, cross-permutation-based relations in IWSS and IWSSr are defined in CPQHFSS. By plugging the different kernels into hyperplane-based wrapper methods of IWSS and IWSSr, all possible cross-permutations of two states per IWSS and IWSSr caused to applying different hybrid FSS on each univariate of transient multivariate excursions (TMEs) to extract OTFs. In the third step, transient analysis based on survived OTFs in the presence of cross-validation scenario evaluates the effectiveness rate of the OTFs in achieving high-performance TA.

The CPQHFSS by interlacing the filter-wraper methods, incremental wrapper mechanisms, and cross-permutation scenario, selects OTFs of TMEs for high-performance TA. According to Fig. 2, each univariates of the transient multivariate trajectories dataset is entered into the filter phase as the first step of CPQHFSS. Then, the obtained filter-based results per univariate are used in the wrapper phase of the CPQHFSS, which consists of four hybrid blocks categorized in twin two-FWBs, which are mounted on dual incremental wrapper mechanisms (IWSS and IWSSr). Overall, in the CPQHFSS, four hybrid blocks formed as IWSS (the left rectangle box of Fig. 2) and IWSSr (the right rectangle box of Fig. 2). The direct relations in each block of IWSS or IWSSr be caused that two FWv states (totally four FWv-states). In CPQHFSS, due to plugging elastic and non-elastic kernels into hyperplane-based approaches situated in the wrapper methods, different cross-permutation hybrid FSS can be considered for two filter-wraper-states of IWSS and IWSSr (See cross-permutation box in Fig. 2). Taking into cognizance the concise explanation of CPQHFSS depicted in Fig. 2, the pseudocode of CPQHFSS is shown in Table 1. As can be seen in Table 1, the main body of CPQHFSS includes filter phase (RR analysis), incremental wrapper mechanisms (IWMs), and cross-permutation function. In the main body of CPQHFSS, symmetric uncertainty values (Line 3) of

![Figure 1](https://example.com/fig1.png)

**FIGURE 1.** The Overall framework of FSS-oriented TA based on CPQHFSS
FIGURE 2. Overall process of CPQHFSS
point features per transient univariate (TU) are calculated. After sorting point features (pfs) of TU\(^k\) based on SU measure (Line 7), these features are entered into IWSS and IWSSr functions (Line 8 and Line 9) to extract preliminary optimal transient features (pOTFs). Each IWM is equipped with dual hyperplane-based classifiers (SVM and TWSVM) to exert the train-test procedures. Also, to find optimal separating hyperplane, different kernels are plugged into SVM and TWSVM classifiers. After selecting pOTFs per TU in a six-states manner by IWMs \(\left[ \begin{array}{c} pOTFs_{IWSS TU}^k \\ \end{array} \right]_{2 \times 3} \) and \(\left[ \begin{array}{c} pOTFs_{IWSS TU}^k \\ \end{array} \right]_{2 \times 3} \), these matrices for each TU are recorded in the structure arrays \(\left[ pOTFs_{IWSS TU}^k, pOTFs_{IWSSr TU}^k \right] \) (Line 10.1109/ACCESS.2022.3166917, IEEE Access

\[ \text{TABLE I} \]

\text{THE PSEUDOCODE OF THE CPQHFSS}

\begin{tabular}{|p{0.9\textwidth}|}
\hline
\textbf{Main body of CPQHFSS} \tabularnewline
\hline
\textbf{Input:} transient univariates (TU) of multivariate trajectory dataset; \(\{TU^k | k=1,2, ..., n\}\). \tabularnewline
\textbf{Output:} union of final optimal transient features \(\cup pf_{Spfs}\). \tabularnewline
(1) \textbf{for} \(k = 1 \text{ to } n\) \tabularnewline
(2) \textbf{for} \(i = 1 \text{ to } m\) // \(m=\text{length (pfs)}, pfs: \text{point features (observed cycles)} \text{in } TU^k\). \tabularnewline
\hspace{1cm} \(SU_TU^k pf_i = \text{calculate relevancy rate (RR) of } pf_i \in TU^k\) \tabularnewline
(4) \textbf{end} \tabularnewline
(5) \textbf{end} \tabularnewline
(6) \textbf{for} \(k = 1 \text{ to } n\) \tabularnewline
(7) \(\text{sort } SU_TU^k pf_i \text{ array in descending manner.}\) \tabularnewline
(8) \(\text{pOTFs}_{IWSSr TU}^k = \text{IWSSr} (SU_TU^k); \) // Recording preliminary optimal transient features (pOTFs) per TU\(^k\) in \(\left[ \begin{array}{c} pOTFs_{IWSSr TU}^k \\ \end{array} \right]_{2 \times 3} \) \tabularnewline
(9) \(\text{pOTFs}_{IWSSr TU}^k = \text{IWSS} (SU_TU^k); \) // Recording preliminary optimal transient features (pOTFs) per TU\(^k\) in \(\left[ \begin{array}{c} pOTFs_{IWSSr TU}^k \\ \end{array} \right]_{2 \times 3} \) \tabularnewline
(10) \textbf{if} \(k = 1\) \tabularnewline
(11) \(\text{pOTFs}_{IWSS TU}^k = \text{struct} (\text{pOTFs}_{IWSSr TU}^k); \) \tabularnewline
(12) \(\text{pOTFs}_{IWSSr TU}^k = \text{struct} (\text{pOTFs}_{IWSSr TU}^k); \) \tabularnewline
(13) \textbf{else} \tabularnewline
(14) \textbf{end} \tabularnewline
(15) \textbf{end} \tabularnewline
(16) \text{pOTFs}_{IWSSr TU}^k = \text{cross-permutation} (\text{pOTFs}_{IWSSr TU}^k, pOTFs_{IWSSr TU}^k); \tabularnewline
(17) \text{pOTFs}_{IWSSr TU}^k = \text{cross-permutation} (\text{pOTFs}_{IWSSr TU}^k, pOTFs_{IWSSr TU}^k); \tabularnewline
(18) \text{pOTFs}_{IWSSr TU}^k = \text{cross-permutation} (\text{pOTFs}_{IWSSr TU}^k, pOTFs_{IWSSr TU}^k); \tabularnewline
\hline
\textbf{Function: IWSSr} \tabularnewline
\hline
(1) \text{Classifier: } \{\text{SVM, TWSVM}\}, \text{Kernelx}\{\text{GRBF, DTW GRBF, REDK}\}, \text{Kernelx}\{\text{GRBF, PolKer, LinKer}\}; \tabularnewline
(2) \text{Sel}\{SU_TU^k\} \{1\}; // \text{first element of SU_TU^k array (feature with highest SU insert in Sel).}\tabularnewline
(3) \text{SVMx}\{\text{Classifier}\{1\} + \text{Kernelx}\{x\}; // \text{x=1 | 2 | 3, Learning model: } \{\text{SVMx}\}\}; \text{Three learning modes.}\tabularnewline
(4) \text{TWS VMx}\{\text{Classifier}\{2\} + \text{Kernelx}\{y\}; // \text{y=1 | 2 | 3, Learning model: } \{\text{TWS VMx}\}\}; \text{Three learning modes.}\tabularnewline
(5) \text{AccSel=} \text{PerEval (Learning Model,} \left[ \begin{array}{c} \text{data} \\ \end{array} \right]_{m \times Sel \text{ with c}}; \) // \(C: \text{[target class]}_{m \times \text{Sel}}; m: \text{number of sample}, \text{PerEval: Performance Evaluation.}\) \tabularnewline
(6) \text{for} \(i = 2 \text{ to length(Sel)}\) \tabularnewline
\hspace{1cm} \text{OptFex=} \emptyset \; \tabularnewline
(7) \text{for} \(r = 1 \text{ to length(Sel)}\) \tabularnewline
\hspace{1cm} \text{Selpred=} \text{updateSelx} (Sel, \text{swap(Sel} \{i\}, \text{ Selpred} \{i\}); \tabularnewline
\hspace{1cm} \text{AccSelpred=} \text{PerEval (Learning Model,} \left[ \begin{array}{c} \text{data} \\ \end{array} \right]_{m \times Selpred \text{ with c}}; \) \tabularnewline
(11) \text{if} (\text{AccSelpred} < \text{AccSel}) \tabularnewline\hspace{1cm} \text{OptFex=} \text{swap (Sel} \{i\}, \text{ Selpred} \{i\}); \tabularnewline\hspace{1cm} \text{AccSel=} \text{AccSelpred}; \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{Selpred=} \text{updateSelx (Sel, Add)\{Selpred\} \{i\}; \tabularnewline\hspace{1cm} \text{AccSelpred=} \text{PerEval (Learning Model,} \left[ \begin{array}{c} \text{data} \\ \end{array} \right]_{m \times Selpred \text{ with c}}; \) \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{AccSelpred=} \text{PerEval (Learning Model,} \left[ \begin{array}{c} \text{data} \\ \end{array} \right]_{m \times Selpred \text{ with c}}; \) \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{AccSelpred=} \text{PerEval (Learning Model,} \left[ \begin{array}{c} \text{data} \\ \end{array} \right]_{m \times Selpred \text{ with c}}; \) \tabularnewline\hspace{1cm} \text{end} \tabularnewline\hspace{1cm} \text{return Sel}; // Sel returned as matrix \(\left[ \begin{array}{c} \text{pOTFs}_{IWSSr TU}^k \\ \end{array} \right]_{2 \times 3}\), first row of matrix related to selected \{Sel\} by SVMx}\{\text{Kernelx}\} and second row of matrix related to selected \{Sel\} by TWS VMx}\{\text{Kernelx}\}. \tabularnewline\hline
\end{tabular}
Hence, the complexity of worst case, the complexity of IWSS and IWSSr is based learning methods. By analyzing these main functions, CPQHFSS for readers at a glance. The complexity of pseudocode of CPQHFSS, we present the complexity of besides the above-mentioned describing the main body of III-C.

\[ f = \text{permutation function to extract final OTFs (U)} \]

Finally, structure arrays are entered into the cross-permutation function to extract final OTFs (fOTFs) per TU (Line 16). After conducting union-intersection operations on subsets of optimal features based on cross-permutation scenario, the union of fOTFs (UfOTFs) is obtained (Line 17). To better understand the details of the pseudocode of CPQHFSS, we elaborate on the triple components of it (filter, IWMs, and cross-permutation) in Sections III-A to III-C.

Besides the above-mentioned describing the main body of pseudocode of CPQHFSS, we present the complexity of CPQHFSS for readers at a glance. The complexity of CPQHFSS is related to IWMs accompanied by hyperplane-based learning methods. By analyzing these main functions, we can approximate the complexity of CPQHFSS. 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 \( O(2x(n/2)^3) \), respectively [25]. Hence, the complexity of \( \text{SVM}_{\text{IWSS}}^{\text{TWSVM}} \) is \( O(\max\{(n/n^3), (n^2x2x(n/2)^3)) \) complexity. Since the complexity of the SVM is 4 times larger than of the TWSVM, the complexity of \( \text{SVM}_{\text{IWSS}}^{\text{TWSVM}} \) and \( \text{SVM}_{\text{IWSSr}}^{\text{TWSVM}} \) will be equal to \( O(n^3) \) and \( O(n^2x3) \), respectively. On the other hand, the experiment results of Reference [24] show the fact that the complexity of IWSSr is near to IWSS when the number of variables to be selected is a very small number of wrapper evaluations. Consequently, the CPQHFSS has \( O(2x3^2) \) complexity.

### A. INCREMENTAL WRAPPER MECHANISMS (IWMs)

#### 1) IWSS

The IWSS mechanism [26] is utilized in the first set of twin FWBs (IWSSFWBs) in CPQHFSS as IWM. How to navigate in the incremental process of IWSS to select optimal features depends on the results of the embedded filter and wrapper method. As the preliminary step of IWSS, by conducting the filter method on feature set based on relevancy ratio (RR), the features are sorted based on

|TABLE I (CONTINUED.)|
|---|
|THE PSEUDOCODE OF THE CPQHFSS|

Function: IWSS

1. \( \text{Sel}^{\text{WPS}\text{TU}} \{1\} \); then \( \text{Sel}^{\text{WPS}\text{TU}} \{1\} \);
2. \( \text{AccSel} = \text{PerEval} (\text{Learning Model}, [\text{data}]_{\text{m} \times \text{Sel with C}}) \);
3. for \( i = 2 \) to length (\( \text{WPS}\text{TU}^3 \))
4. \( \text{Sel}^{\text{WPS}\text{TU}^3} \{j\} \);
5. \( \text{AccSel}^{\text{WPS}\text{TU}^3} = \text{PerEval} (\text{Learning Model}, [\text{data}]_{\text{m} \times \text{Sel with C}}) \);
6. if \( (\text{AccSel}^{\text{WPS}\text{TU}^3} = \text{AccSel}) \)
7. \( \text{add} (\text{Sel}, \text{Spfs}^3 \{1\}) \);
8. \( \text{AccSel} = \text{AccSel}^{\text{WPS}\text{TU}^3} \);
9. end
10. end
11. return \( \text{Sel} \); // Sel returned as matrix \( [\text{OTFs}]_{\text{TU}^3} \), first row of matrix related to selected \{Sel\} by SVMKernels and second row of matrix related to selected \{Sel\} by TWSVMKernels.

Function: Cross-Permutation

1. for \( k = 1 \) to \( n \)
2. \( \text{IWSSmAt} = \text{fOTF}\text{IWSS}^\text{TU}^3 \{k\}, \text{fOTF}\text{TU}; // \text{insert k}^{th} 6\text{-element matrix} (\text{[pOTF}\text{IWSS}^\text{TU}^3 \{k\}_{\text{TU}}\text{mat}) \) of struct \( \text{fOTF}\text{IWSS}^\text{TU} \) in \( \text{IWSSmAt} \).
3. \( \text{IWSSmAt} = \text{fOTF}\text{IWSS}^\text{TU}^3 \{k\}, \text{fOTF}\text{TU}; // \text{insert k}^{th} 6\text{-element matrix} (\text{[pOTF}\text{IWSS}^\text{TU}^3 \{k\}_{\text{TU}}\text{mat}) \) of struct \( \text{fOTF}\text{IWSS}^\text{TU} \) in \( \text{IWSSmAt} \).
4. // first row of matrix \( [\text{IWSS}/\text{IWSSmat}] \) \( k \) \( \times 3 \) (elements \( [1, 3, 5] \)) related to selected \{Sel\} by SVMKernels.
5. // second row of matrix \( [\text{IWSS}/\text{IWSSmat}] \) \( 2 \times 3 \) (elements \( [2, 4, 6] \)) related to selected \{Sel\} by TWSVMKernels.
6. // Permutations of IWSS-based learning models (9 permutations (LP1: LP9))
7. \( \text{LP1} = \text{IWSSmAt} (1) U \text{IWSSmAt} (2); \text{LP2} = \text{IWSSmAt} (1) U \text{IWSSmAt} (4); \text{LP3} = \text{IWSSmAt} (1) U \text{IWSSmAt} (6); \)
8. \( \text{LP4} = \text{IWSSmAt} (3) U \text{IWSSmAt} (2); \text{LP5} = \text{IWSSmAt} (3) U \text{IWSSmAt} (4); \text{LP6} = \text{IWSSmAt} (3) U \text{IWSSmAt} (6); \)
9. \( \text{LP7} = \text{IWSSmAt} (5) U \text{IWSSmAt} (2); \text{LP8} = \text{IWSSmAt} (5) U \text{IWSSmAt} (4); \text{LP9} = \text{IWSSmAt} (5) U \text{IWSSmAt} (6); \)
10. // Permutations of IWSSr-based learning models (9 permutations (RP1: RP9))
11. \( \text{RP1} = \text{IWSSmAt} (1) U \text{IWSSmAt} (2); \text{RP2} = \text{IWSSmAt} (1) U \text{IWSSmAt} (4); \text{RP3} = \text{IWSSmAt} (1) U \text{IWSSmAt} (6); \)
12. \( \text{RP4} = \text{IWSSmAt} (3) U \text{IWSSmAt} (2); \text{RP5} = \text{IWSSmAt} (3) U \text{IWSSmAt} (4); \text{RP6} = \text{IWSSmAt} (3) U \text{IWSSmAt} (6); \)
13. \( \text{RP7} = \text{IWSSmAt} (5) U \text{IWSSmAt} (2); \text{RP8} = \text{IWSSmAt} (5) U \text{IWSSmAt} (4); \text{RP9} = \text{IWSSmAt} (5) U \text{IWSSmAt} (6); \)
14. // Cross relations.
15. \( \text{C1} = \text{U} \{ \text{LP1} \cap \text{RP1}; \text{RP9} \}; \text{C2} = \text{U} \{ \text{LP2} \cap \text{RP1}; \text{RP9} \}; \text{C3} = \text{U} \{ \text{LP3} \cap \text{RP1}; \text{RP9} \}; \)
16. \( \text{C4} = \text{U} \{ \text{LP4} \cap \text{RP1}; \text{RP9} \}; \text{C5} = \text{U} \{ \text{LP5} \cap \text{RP1}; \text{RP9} \}; \text{C6} = \text{U} \{ \text{LP6} \cap \text{RP1}; \text{RP9} \}; \)
17. \( \text{C7} = \text{U} \{ \text{LP7} \cap \text{RP1}; \text{RP9} \}; \text{C8} = \text{U} \{ \text{LP8} \cap \text{RP1}; \text{RP9} \}; \text{C9} = \text{U} \{ \text{LP9} \cap \text{RP1}; \text{RP9} \}; \)
18. \( \text{fOTF}\text{IWSS}^{\text{IWSSmat}} \{C1 \cap C2 \cap \ldots \cap C9\}; \)
19. \( \text{end} \)
RR values in descending manner. Then, for completing the first increment of IWSS, the feature inserted into the first position of the sorted array ($f_{h1}$: feature with highest RR) is fed to the classification learner (CL), and then $f_{h1}$ with the prediction accuracy ($\text{Acc}(f_{h1})$) is recorded in the candidate features subset (CFS) based on the TTP. In the next increment, the feature with second-highest RR ($f_{h2}$) is added to the CFS, and the updated CFS-based learning model reports the $\text{Acc}(f_{h1}, f_{h2})$. If the classification performance of CFS including $f_{h1}$ and $f_{h2}$ is higher than the performance of $f_{h1}$, the third increase (adding $f_{h3}$) is accompanied by $f_{h1}$ & $f_{h2}$; otherwise, $f_{h2}$ is deleted from CFS, and $f_{h3}$ is added to the CFS and placed next to $f_{h1}$. Fig. 3 shows how to select OTFs by the IWSS in the form of a numerical example.

2) IWSSr

In the second set of twin 2FWBs, filter-wrapper blocks are mounted on IWSSr [24] algorithm (IWSSr 2FWBs) as IWM. In IWSSr, similar to the dependence of the IWSS on filter and wrapper method results, based on sorted RR values of features, in the first increment, $f_{h1}$ is added to CFS, then CL trained by $f_{h1}$ and $\text{Acc}(f_{h1})$ is recorded. In the second increment, $f_{h2}$ is added to the preceding CFS in two modes. In the first mode, $f_{h1}$ is replaced by $f_{h2}$ (only $f_{h2}$ added to CFS) and in the second mode, $f_{h1}$ and $f_{h2}$ are added to CL together. Now, in the second increment, $\text{Acc}(f_{h1})$ and $\text{Acc}(f_{h1}, f_{h2})$ are obtained. Fig. 4 shows the process of IWSSr which the third increment starts from node 3 (create node 4 to 6).

B. TWIN TWO FILTER-WRAPPER BLOCKS (2FWBs)

Twin 2FWBs in CPOHSS refer to applying two sets of 2FWBs in the presence of IWMs which are called

IWSS$^{2FWBs}$ and IWSS$^{2FWBs}_{r}$. The 2FWBs include frozen information theory concept (filter-fixed) and unfixed machine learning classifier (wrapper-varied). Filter-related model of 2FWBs including the relevancy ratio (RR) [27] in the first-second hybrid block (HB$^{1-2}$) of IWSS$^{2FWBs}$ ($RR_{IWSS}^{2FWBs}$) and IWSS$^{2FWBs}_{r}$ ($RR_{IWSS}^{2FWBs}_{r}$). Wrapper-related models including support vector machine in the HB$^{1}$ of IWSS$^{2FWBs}$ ($\text{SVM}_{IWSS}^{2FWBs}$) and IWSS$^{2FWBs}_{r}$ ($\text{SVM}_{IWSS}^{2FWBs}_{r}$) and twin support vector machine (TWSVM) in HB$^{2}$ of IWSS$^{2FWBs}$ ($\text{TWSVM}_{IWSS}^{2FWBs}$) and IWSS$^{2FWBs}_{r}$ ($\text{TWSVM}_{IWSS}^{2FWBs}_{r}$). Generally, in respect of twin 2FWBs, one set of 2FWBs is situated in IWSS formed as $\text{RR}_{IWSS}^{2FWBs}$ and $\text{TWSVM}_{IWSS}^{2FWBs}$. Also, another set of 2FWBs is associated with IWSSr raised as $\text{RR}_{IWSS}^{2FWBs}_{r}$ and $\text{TWSVM}_{IWSS}^{2FWBs}_{r}$. The detailed descriptions of the filter and wrapper models of 2FWBs are discussed in the following subsections.

1) Filter-Fixed methods in 2FWBs

Relevancy ratio (RR): The symmetric uncertainty (SU) is considered as the HB$^{1-2}$-specific filter index in IWSS$^{2FWBs}$ and IWSS$^{2FWBs}_{r}$ which triangulated the entropy, conditional entropy, and mutual information (MI) to measure the relevancy rate between features of transient univariates and class label. The SU index is calculated as:
the data $f_{i}^{TU}$ is employed for mapping the data 

$$SU(f_{i}^{TU}, C) = 2 \frac{MI(f_{i}^{TU}; C)}{H(f_{i}^{TU}) + H(C)}$$

(1)

Where $f_{i}^{TU}$ represents $i$th features of $k$th transient univariate, and $C$ is the class label of transient samples. In (1), the entropy $H(Z)$ is defined as:

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

(2)

Where $Z$ is a discrete random variable and $p(z) = \Pr[Z = z]$ is a probability density function. Mutual information (MI) in (1) is defined as:

$$MI(f_{i}^{TU}; C) = H(f_{i}^{TU}) - H(f_{i}^{TU}| C)$$

(3)

Where $H(f_{i}^{TU}| C)$ in (3) is called conditional entropy as follow:

$$H(f_{i}^{TU}| C) = -\sum_{x \in f_{i}^{TU}} \sum_{c \in C} p(x, c) \log p(x| c)$$

(4)

2) Wrapper methods in 2FWBs

(a) Support vector machine (SVM)

SVM in [28] introduced as a supervised learning model that draws hyperplane in feature space for classifying binary or multi-class data in the form of the linear SVM (hard margin or soft margin approach) and nonlinear SVM (kernel-based approach). Regardless of the different factors that affected the SVM formula, SVM aims a low structural risk without overfitting data to achieve high accuracy in train-test procedures. For example, achieving such a goal in the presence of data that has a nonlinear decision boundary (e.g., PMU-gathered HDTF), necessitates plugging the kernel trick into SVM computations as follows:

$$a^* = \arg \min_{a} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_{i} \alpha_{j} y_{i} y_{j} K(x_{i}, x_{j}) - \sum_{k=1}^{l} \alpha_{k};$$

$$0 \leq \alpha_{k} \leq C, \quad \sum_{j=1}^{l} \alpha_{j} y_{j} = 0, \quad i, j = 1, \ldots, l$$

(5)

Where $K(x_{i}, x_{j})$ in (5) is employed for mapping the data from the main space to the new space (high dimensional space) so that in the new space the data are linearly separable. The maximum-margin separating hyperplane in feature space is solved by (6):

$$f(x) = \text{sgn} \left( \sum_{i \in S} \alpha_{i} y_{i} K(x_{i}, x) + b \right);$$

$$b = \frac{1}{s} \sum_{i \in S} y_{i} - \sum_{j \in S} \alpha_{j} y_{j} K(x_{j}, x_{i})$$

(6)

(b) Twin support vector machine (TWSVM)

The standard SVM is formulated based on finding the middle boundary (maximum margin) between two parallel planes with the maximum distance from each of the existing classes. SVM geometry space can be reshaped by cross planes in which each plane could nearest distance to the samples of one class and farthest from the samples of the other class. This idea was raised as the generalized proximal eigenvalue support vector machine (GEPSVM) [28]. In another effort by [29], the spirit of the GEPSVM was kept into a new skeleton (new formulation) was termed TWSVM. In TWSVM, cross planes are obtained by solving the following optimization problems:

$$\min_{w_1, h_1, \xi_2} \frac{1}{2} \| Pw_1 + e_1 b_1 \|_2^2 + c_1 e_1^T \xi_2$$

s.t. $-(e_1 + e_2 b_2) + \xi_2 \geq e_2, \quad \xi_2 \geq 0$

(7)

$$\min_{w_2, b_2, \xi_1} \frac{1}{2} \| Qw_2 + e_2 b_2 \|_2^2 + c_2 e_2^T \xi_1$$

s.t. $(Pw_2 + e_1 b_2) + \xi_1 \geq e_1, \quad \xi_1 \geq 0$

(8)

Where $c_1$, $c_2$, $e_1$, and $e_2$ are vectors with a value of 1 and a proper dimension. By calculating the Lagrangian function for (7) and (8), the Karush–Kuhn–Tucker (KKT) equations are formed. By placing the KKT terms and relations in the Lagrangian function for each of the relations (7) and (8), the dual optimal relations are obtained according to the following relations:

$$\text{dual TWSVM}^1: \max \{e_1^T \alpha - \frac{1}{2} \alpha^T G(H^T H)^{-1} G^T \alpha\}$$

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

(9)

(10)

Based on the dual optimization problems, the value $\alpha$ and $\psi$ is obtained via quadratic programming and by placing these values in the KKT relations, the values $[w^{(1)}, b^{(1)}]$ and $[w^{(2)}, b^{(2)}]$ related to hyperplanes of the binary-class task are obtained:

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

(11)

Finally, the class label of the unseen point $x \in \mathbb{R}^n$ is determined from the plane that is close to this point:

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

(12)

Based on the above-mentioned principle of TWSVM, the nonlinear classification of HDTF can be considered via kernel-based cross planes [29]:

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

(13)

Where $C^T = [A B]^T$ and $K$ denote the kernel. The vector $[u^{(1)} y^{(1)}]^T$ and $[u^{(2)} y^{(2)}]^T$ are obtained by solving the following optimization problem:
\[ K_{\text{TWSVM}}^1 : \min_{u^{(1)}, b^{(1)}, q} \frac{1}{2} \| (A, C^T) u^{(1)} + e_b^{(1)} \|^2 + c_1 e_2^T q \]  
\[ \text{s.t.} \ - (K(B, C^T) u^{(1)} + e_b^{(1)}) + q \geq e_2, \ q \geq 0 \]  
\[ K_{\text{TWSVM}}^2 : \min_{u^{(2)}, b^{(2)}, q} \frac{1}{2} \| (K(B, C^T) u^{(2)} + e_b^{(2)}) \|^2 + c_2 e_2^T q \]  
\[ \text{s.t.} \ (K(A, C^T) u^{(2)} + e_b^{(2)}) + q \geq e_1, \ q \geq 0 \]

**C. CROSS-PERMUTATION SCENARIO BASED ON WRAPPER KERNELS**

The cross-permutation scenario in CPQHFFS is conducted based on plugged triple kernel into wrapper approaches of twin 2FWBs. The kernels situated in wrapper approaches of IWSS^{2FWBs} and IWSSr^{2FWBs} are categorized into two types: (1) non-elastic kernel: linear kernel (LinKer) [31], polynomial kernel (PolKer) [31], and standard Gaussian radial basis function (GRBF) [28], and (2) elastic kernels: dynamic time warping in GRBF (DTW/GRBF) [32] and recursive edit distance kernel (REDK) [33]. The definitions per kernel are summarized below:

(a) LinKer: LineKer is calculated based on the inner product plus constant c, which is considered as the simplest kernel function:
\[ K(x, x') = x^T x' + c \]  
(b) PolKer: The degree-based variant of LinKer (the d value more than 1) is known as PolKer, which is defined as follows:
\[ K(x, x') = (a x^T x' + c)^d \]

(c) \( ^5 \text{GRBF} \): \( ^5 \text{GRBF} \) is known as a non-elastic kernel due to linear alignment (point to point) in pattern matching in feature space. The \( ^5 \text{GRBF} \) for using as \( K(x, x') \) in (5) and (13) is defined as follows:
\[ K(x, x') = \exp \left( -\frac{||x - x'||^2}{2\sigma^2} \right) \]

The distance between two trajectories is calculated based on squared Euclidean distance denoted by \( \|x - x'\|^2 \) in (18).

(d) DTW/GRBF: The elastic behavior of DTW motivates to replace Euclidean distance with DTW distance in \( ^5 \text{GRBF} \) to have a robust and non-linear alignment in feature space. The DTW distance between two-time series is calculated by (19):
\[ \text{distance}^{\text{DTW}}(A^p_1, B^q_1) = d(a(p), b(q)) + \min \left( \begin{array}{l} \text{distance}^{\text{DTW}}(A^p_1, B^q_1) \\ \text{distance}^{\text{DTW}}(A^{p-1}_1, B^q_1) \\ \text{distance}^{\text{DTW}}(A^p_1, B^{q-1}_1) \end{array} \right) \]

According to (19), \( K(x, x') \) can be defined as the DTW-based elastic kernel as follow:
\[ K(x, x') = \exp \left( -\frac{\text{distance}^{\text{DTW}}(A^p_1, B^q_1)^2}{2\sigma^2} \right) \]

(e) REDK: In [33], constructing the kernel based on the aggregation of scores recursively caused that the REDK is introduced as an efficient elastic similarity measure against classical elastic distances. According to (21), if for any pair of trajectory, such following equation is satisfied, the function \( \langle \cdot, \cdot \rangle: U \times U \rightarrow R \) termed as REDK:
\[ \langle A^p, B^q \rangle = \sum \left( \begin{array}{l} \langle A^{p-1}, B^q \rangle f(\Gamma(A(p) \rightarrow A)) \\ \langle A^p, B^{q-1} \rangle f(\Gamma(A(p) \rightarrow B(q))) \end{array} \right) \]

Where \( U = \{A^p | p \in \mathbb{N}\} \). \( A^p \) (or \( B^q \)) is the trajectory with a discrete index varying between 1 and \( p \) (or \( q \)). Also, \( \Gamma(h) \) is the cost function for edit operation.

After applying different kernels in the hyperplane classifiers of wrapper phase (\( ^5 \text{GRBF}, \text{DTW/GRBF}, \) and REDK in SVM; \( ^5 \text{GRBF}, \text{LinKer}, \) and PolKer in TWSVM) in the form of proposed IWMs (\( K_{\text{SU}} \) and \( K_{\text{SU}} \) ) in IWSS^{2FWBs} and IWSSr^{2FWBs}, the union of the selected preliminary OTFs (pOTFs) per the state of IWSS^{2FWBs} is recorded (See Fig. 2, left panel (LLP and RLP) of the cross-permutation box). In terms of IWSS^{2FWBs}, the mentioned scenario is conducted to record union results per permutation in IWSSr^{2FWBs}, which is shown in the right panel (LRP and RRP) of the permutation box. Next, based on the cross-manner scenario, each left union is linked with right unions (9 links are established; e.g., 9 black links, 9 blue links, and so on). Then, the intersection of both sides of the link in each colored 9 links (e.g., the intersection of both sides of the first black link) is calculated. After calculating the intersection of both sides of all colored links (e.g., both sides of the second black link to both sides of the ninth black link), the union of sets is recorded (See Fig. 2), union sign in the polygonal black side). Such mechanism (intersection and union) is conducted on other colored 9 links (9 blue links, 9 yellow links, and so on). Finally, the intersection of the obtained sets in polygonal sides is considered as univariate-specific final OTFs (intersection of polygonal black side, polygonal brown side, and so on).
IV. EXPERIMENTAL DESIGN

A. TRANSIENT DATASET CONSTRUCTION

As can be seen in Fig. 1, transient dataset construction containing multivariate trajectories is considered as the first step of the proposed framework for FSS-based TSA. In this step, we executed contingency simulation based on the two-step transient dataset generation workflow (TDGW) proposed in [34], which is shown in Fig. 5. In TDGW, first, transient responses are extracted from output channels of basic features (OCBF-X; X indicated basic features namely bus voltages (VOLT), voltage phase angle (VANGLE), machine active power (PELEC), machine reactive power (QELEC), and reactive power consumption (QLOAD)). The application program interface (API) functions of the SIEMENS power system simulator for engineering (PSS/E) [35] in the form of Python-based automation file are used for exerting contingency simulation on the New England test system- New York power system (NETS-NYPS) (See Fig. 6) [36]. The generated transient samples are derived by substation outages, generator outages, and line outages where the fault duration time is set to 0.23 seconds (the time step is 0.0167 seconds). Also, the fault clearing time is set after the end of fault duration time. As an important uncertainty parameter in generating transient samples, the convert load (CONL) API is added to the PSS/E-Python automation file to regard different load characteristics which the percent of active and reactive power load to be converted to the constant current and constant admittance load characteristics [34, 36]. In the second step of TDGW, a set of Matlab-based commands cause adding factors to OCBF-X which leads to the defined 28 univariates trajectory features listed in Table 2 [37, 38, 39]. After conducting TDGW, the transient dataset contains 800 (No. transient samples) × 28 (No. features) × 9 (No. observed cycles). For example, some stable and unstable samples streamed based on \( F_{14} \) trajectory (The proportion of total QLOAD to total QELEC) is shown in Fig. 7.

B. FINAL OTFs (fOTFs) SET PER TRANSIENT UNIVARIATE (TU)

Extracting the transient univariate-specific final OTFs set called \( \text{fOTFs}_{TU} \) (namely \( \text{fOTFs}_{TU1} \) to \( \text{fOTFs}_{TU28} \)) by applying CPQHFSS on each univariate (TU1 to TU28) of transient multivariate trajectory dataset (TMTD) is elaborated in this section. According of Fig. 2, in the first step of CPQHFSS, each univariate of TMTD is entered to
the filter-fixed phase of twin 2FWBs. According to (1), RR of point features per TU of TMTD is calculated based on SU measure. According to what mentioned about the filter method in Section 2.B.1, for example, the obtained RR of \( pf_6 \) of TU \( 6 \) based on SU is shown in Fig. 8. As can be seen in Fig. 8, based on SU values of \( pf_6 \), \{pf\} is ranked as a high SU feature, and \{pf\} is considered as the low SU \( pf \). Next, by sorting \( pf_6 \) based on SU values in descending manner (\( pf_6^{\text{SU}} \)), the order in which the \( pf_6^{\text{SU}} \) enter the wrapper phase in twin 2FWBs is specified (See Table 3; sixth row). For more clarity, Table 3 show the \( pf_1^{\text{SU}} \) to \( pf_6^{\text{SU}} \) based on SU values.

### TABLE II

| Transient Multivariate Time Series Features (28-Variate) |
|--------------------------------------------------------|
| Math formula                                           |
| \( F^1_m = \max(\frac{\text{PELEC}_i}{\text{P}_{\max}}) \) |
| \( F^2_m = \text{Var}(\frac{\text{PELEC}_i}{\text{P}_{\max}}) \) |
| \( F^3_m = \max(\frac{\text{QELEC}_i}{\text{Q}_{\max}}) \) |
| \( F^4_m = \text{Min}(\frac{\text{QELEC}_i}{\text{Q}_{\max}}) \) |
| \( F^5_m = \text{Var}(\frac{\text{QELEC}_i}{\text{Q}_{\max}}) \) |
| \( F^6_m = \max(\sqrt[28]{\text{VOLT}_i}) \) |
| \( F^7_m = \text{Var}(\sqrt[28]{\text{VOLT}_i}) \) |
| \( F^8_m = \max(\sqrt[28]{\text{VANGLE}_{\text{slack bus}}}) \) |
| \( F^9_m = \text{Min}(\sqrt[28]{\text{VANGLE}_{\text{slack bus}}}) \) |
| \( F^{10}_m = \text{Var}(\sqrt[28]{\text{VANGLE}_{\text{slack bus}}}) \) |
| \( F^{11}_m = \max(\text{abs}(\sqrt[28]{\text{VANGLE}_{i,j}} - \sqrt[28]{\text{VANGLE}_{i,j}})) \) |
| \( F^{12}_m = \text{Mean}(\text{abs}(\sqrt[28]{\text{VANGLE}_{i,j}} - \sqrt[28]{\text{VANGLE}_{i,j}})) \) |
| \( F^{13}_m = \text{Var}(\text{abs}(\sqrt[28]{\text{VANGLE}_{i,j}} - \sqrt[28]{\text{VANGLE}_{i,j}})) \) |
| \( F^{14}_m = \sum_{i=1}^{N_{\text{bus}}} \sum_{j=1}^{N_{\text{bus}}} \text{QLOAD}_i \) |
| \( F^{15}_m = \frac{\sum_{i=1}^{N_{\text{bus}}} \text{QLOAD}_i}{\sum_{i=1}^{N_{\text{bus}}} \text{QELEC}_i} \) |

Symbol: \( t \) = moments in simulation time [1: s], \( N_{\text{bus gen}} \) = number of bus generator in test case, \( \text{PELEC} \) = machine electrical power (pu), \( P_{\max} \) = maximum amount of machine electrical power, \( \text{QELEC} \) = machine reactive power, \( Q_{\max} \) = maximum amount of machine reactive power, \( \text{Qload} \) = reactive power consumption, \( \text{Volt} \) = bus pu voltages, \( N_{\text{bus}} \) = number of buses in test case, \( \text{VANGLE} \) = voltage phase angle, \( \text{Var} \) = variance, \( \text{Max} \) = maximum, \( \text{Min} \) = minimum.
After calculating the SU amount of $p^\text{OTF}_{TU^1}^{1:28}$ and sorting the $pfs$ of $TU^{1:28}$ ($p^\text{OTF}_{TU}^{1:28}$), the $p^\text{OTF}_{TU}^{1:28}$ are fed to the wrapper phase of twin 2FWBs in the form of IWSS$^{2\text{FWBs}}$ and IWSSr$^{2\text{FWBs}}$. In IWSS$^{2\text{FWBs}}$ (the left rectangle box in Fig. 2), the first hybrid block including SVM classifier ($\text{SVM}_{IWSS}^{2\text{FWBs}}$) accompanied with $SGRBF$, $DTW-GRBF$, and REDK kernels. Also, the second hybrid block of IWSS$^{2\text{FWBs}}$ includes the TWSVM classifier ($\text{TWSVM}_{IWSS}^{2\text{FWBs}}$) with LinPer, PolPer, and $SGRBF$ kernels. In terms of IWSSr$^{2\text{FWBs}}$ (the right rectangle box in Fig. 2), the $p^\text{OTF}_{TU}^{1:28}$ entered into $\text{SVM}_{IWSS}^{2\text{FWBs}}$ (kernels: $SGRBF$, $DTW-GRBF$, and REDK) and $\text{TWSVM}_{IWSS}^{2\text{FWBs}}$ (kernels: LinPer, PolPer and $SGRBF$).

After conducting the wrapper phase of CPQHFS in the form IWSS$^{2\text{FWBs}}$ and IWSSr$^{2\text{FWBs}}$, the obtained results are shown in Table 4 to Table 7. Table 4 show the obtained preliminary OTFs ($p^\text{OTFs}$) of $TU^1$ ($p^\text{OTF}_{TU^1}$) based on IWSS-SVM regarding SVM-specific kernels in the form of $p^\text{OTF}_{IWSS-SVM}^{SGRBF}$, $p^\text{OTF}_{IWSS-SVM}^{DTW-GRBF}$, and $p^\text{OTF}_{IWSS-SVM}^{REDK}$ columns. Also, Table 5 show $p^\text{OTF}_{TU^x}$ based on IWSS-TWSVM regarding TWSVM-specific kernels in the form of $p^\text{OTF}_{IWSS-TWSVM}^{LinPer}$, $p^\text{OTF}_{IWSS-TWSVM}^{PolPer}$, and $p^\text{OTF}_{IWSS-TWSVM}^{SGRBF}$ columns. In terms of IWSSr-based results, Table 6 and Table 7, show the $p^\text{OTF}_{TU^x}$ based on IWSSr-SVM ($p^\text{OTF}_{IWSSr-SVM}^{SGRBF}$, $p^\text{OTF}_{IWSSr-SVM}^{DTW-GRBF}$, and $p^\text{OTF}_{IWSSr-SVM}^{REDK}$) and IWSSr-TWSVM ($p^\text{OTF}_{IWSSr-TWSVM}^{LinPer}$, $p^\text{OTF}_{IWSSr-TWSVM}^{PolPer}$, and $p^\text{OTF}_{IWSSr-TWSVM}^{SGRBF}$) respectively. For example, after applying the wrapper phase of CPQHFS on $p^\text{OTF}_{TU^3}$, the IWSS-based $p^\text{OTF}_{TU^3}$ (See third row of Table 4 and Table 5) and IWSSr-based $p^\text{OTF}_{TU^3}$ (See third row of Table 6 and Table 7) are extracted. By applying IWSS-SVM$^{SGRBF}$, IWSS-SVM$^{DTW-GRBF}$, and IWSS-SVM$^{REDK}$ in $p^\text{OTF}_{TU^3}$, the IWSS-SVM$^{SGRBF}$ ($pfs$, $pf$s, $pf$s): 90.24%), $p^\text{OTF}_{IWSS-SVM}^{SGRBF}$ ($pfs$, $pf$s, $pf$s): 90.24%), and $p^\text{OTF}_{IWSS-SVM}^{REDK}$ ($pfs$, $pf$s, $pf$s): 90.24%) are obtained, respectively. Also, by conducting IWSS-TWSVM$^{LinPer}$, IWSS-TWSVM$^{PolPer}$, and IWSS-TWSVM$^{SGRBF}$ on $p^\text{OTF}_{TU^3}$, the $p^\text{OTF}_{IWSS-TWSVM}^{LinPer}$ ($pfs$, $pf$s, $pf$s): 80.48%), $p^\text{OTF}_{IWSS-TWSVM}^{PolPer}$ ($pfs$, $pf$s, $pf$s): 80.48%), and $p^\text{OTF}_{IWSS-TWSVM}^{SGRBF}$ ($pfs$, $pf$s, $pf$s): 90.24%) are obtained, respectively. In the case of $p^\text{OTF}_{TU^3}$-specific $p$OTFs based on IWSSr-oriented wrapper phase, the $p^\text{OTF}_{IWSSr-SVM}^{SGRBF}$ ($pfs$, $pf$s, $pf$s): 90.24%), $p^\text{OTF}_{IWSSr-SVM}^{DTW-GRBF}$ ($pfs$, $pf$s, $pf$s): 92.68%), and $p^\text{OTF}_{IWSSr-SVM}^{REDK}$ ($pfs$, $pf$s, $pf$s): 92.68%) are also obtained. Also, the $p^\text{OTF}_{IWSSr-TWSVM}^{SGRBF}$ ($pfs$, $pf$s, $pf$s): 85.36%), $p^\text{OTF}_{IWSSr-TWSVM}^{PolPer}$ ($pfs$, $pf$s, $pf$s): 90.24%) and $p^\text{OTF}_{IWSSr-TWSVM}^{SGRBF}$ ($pfs$, $pf$s, $pf$s): 90.24%).

Finally, based on IWSS-SVM$^{SGRBF}$-based $p^\text{OTF}_{TU^3}$ are selected in IWSS iterations. Also, the Acc metric (23) measured the performance of SVM$^{IWSS}$ and SVM$^{IWSSr}$. Furthermore, the fine-tuning on learning parameters ($C$ and $\sigma$) in each iteration of IWMs is considered for TTP which for the SVM and TWSVM classifiers and their plugged kernels as (24). In the case of IWSS-SVM$^{SGRBF}$ (See Fig. 9), in each iteration, the maximum value of the Acc (obtained by optimal pair of learning parameters) is recorded. As can be seen in Fig. 9, the $p^\text{OTF}_{TU^3}$ based on IWSS-SVM$^{SGRBF}$ is ($pfs$, $pf$s, $pf$s), which Acc variations related to optimal iteration (node 5: green-face) is depicted in 3-D plot.

\[
\text{Accuracy}(\text{Acc}) = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (23)
\]

\[
\begin{align*}
P: & \text{ stable sample, } T: \text{ predicted correctly} \\
N: & \text{ unstable sample, } F: \text{ predicted incorrectly}
\end{align*}
\]

\[
\text{SVM (plugged kernel)}
\begin{align*}
C_2 = & \left| i = 0,1,...,5 \right| \\
\sigma_2 = & \left| j = -3,-2,...,4 \right|
\end{align*}
\]

\[
\text{TWSVM (plugged kernel)}
\begin{align*}
C_2 = & \left| i = 0,1,...,5 \right| \\
\sigma_2 = & \left| j = -3,-2,...,4 \right| \\
\sigma_2 = & \left| j = 0,1,...,4 \right| \quad (\text{PolPer})
\end{align*}
\]

After selecting SVM$^{3\text{ker}}$-based three packages-$p$OTFs and TWSVM$^{3\text{ker}}$-based three packages-$p$OTFs (totally six packages-$p$OTFs-IWSS) and SVM$^{IWSSr3\text{ker}}$-based three packages-$p$OTFs and TWSVM$^{IWSSr3\text{ker}}$-based three packages-$p$OTFs (totally six packages-$p$OTFs-IWSSr per $TU^3$, as the final phase of CPQHFS, which is shown in Fig. 2, the cross-permutation scenario is conducted on obtained 12 packages of $p$OTFs per $TU^3$ for extracting $f$OTFs$^{TU1:TU28}$. Finally, based on $f$OTFs$^{TU1:TU28}$, the union of $f$OTFs$^{TU1:TU28}$ ($U$) obtained as the final optimal transient feature set for use in TSP. The obtained $f$OTFs$^{TU1:TU28}$ based on the cross-permutation phase of CPQHFS is shown in Table 8. Table 8, the details of obtaining the optimal transient features of $TU^3$ ($f$OTFs$^{TU}$) based on cross-permutation are elaborated.
TABLE III
FILTER-BASED RANKING OF POINT FEATURES OF UNIVARIATES

| Input     | Sym TU* based on SU | Input     | Sym TU* based on SU |
|-----------|---------------------|-----------|---------------------|
| TU^1      | ['pf6', 'pf3', 'pf4', 'pf5', 'pf7', 'pf9', 'pf8', 'pf1'] | TU^15     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^2      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^16     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^3      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^17     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^4      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^18     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^5      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^19     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^6      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^20     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^7      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^21     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^8      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^22     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^9      | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^23     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^10     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^24     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^11     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^25     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^12     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^26     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^13     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^27     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |
| TU^14     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] | TU^28     | ['pf1', 'pf2', 'pf3', 'pf4', 'pf5', 'pf6', 'pf7', 'pf8'] |

Symbol: pf: point feature; TU: transient univariate; [pf1, pf2]: sorted pf of TU. *Sym TU*: Sorted pf of TU based on SU

FIGURE 9. Structure of IWSS tree in selecting IWSS-SVM*GRBF based Sym TU* regarding Acc variations in optimal node (node 5)
### TABLE IV

| Input | $p^{1\text{st}}$ | $p^{2\text{nd}}$ | $p^{3\text{rd}}$ | $p^{4\text{th}}$ | TSP Acc | $p^{1\text{st}}$ | $p^{2\text{nd}}$ | $p^{3\text{rd}}$ | $p^{4\text{th}}$ | TSP Acc | $p^{1\text{st}}$ | $p^{2\text{nd}}$ | $p^{3\text{rd}}$ | $p^{4\text{th}}$ | TSP Acc |
|-------|-----------------|-----------------|-----------------|-----------------|--------|-----------------|-----------------|-----------------|-----------------|--------|-----------------|-----------------|-----------------|-----------------|--------|-----------------|-----------------|
| $p^{1\text{st}}$ | 4.29% | 20.89% | 78.04% | 80.48% | $p^{2\text{nd}}$ | 8.75% | 70.36% | 80.48% | 80.48% | $p^{3\text{rd}}$ | 90.24% | 90.24% | 90.24% | 90.24% | $p^{4\text{th}}$ | 90.24% | 90.24% | 90.24% | $p^{5\text{th}}$ | 90.24% | 90.24% | 90.24% | 90.24% |
| $p^{1\text{st}}$ | 4.29% | 20.89% | 78.04% | 80.48% | $p^{2\text{nd}}$ | 8.75% | 70.36% | 80.48% | 80.48% | $p^{3\text{rd}}$ | 90.24% | 90.24% | 90.24% | 90.24% | $p^{4\text{th}}$ | 90.24% | 90.24% | 90.24% | $p^{5\text{th}}$ | 90.24% | 90.24% | 90.24% | 90.24% |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
### TABLE VI

RESULTS OF WRAPPER PHASE BASED ON IWSSR-SVM REGARDING DIFFERENT KERNELS PER TRANSIENT UNIVARIATE

| Input | \(p_{OPT}^{IWSS-SYMREF}\) | \(p_{OPT}^{IWSS-SYMGRBF}\) | \(p_{OPT}^{IWSS-SYMPREDK}\) |
|-------|-----------------|-----------------|-----------------|
| SpfsTu | \([pf_1]: 78.04\%\) | \([pf_1]: 80.48\%\) | \([pf_1]: 87.80\%\) |
| SpfsTu | \([pf_1]: 73.17\%\) | \([pf_1]: 85.36\%\) | \([pf_1]: 90.24\%\) |
| SpfsTu | \([pf_1]: 85.36\%\) | \([pf_1]: 90.24\%\) | \([pf_1]: 95.12\%\) |
| SpfsTu | \([pf_1]: 73.17\%\) | \([pf_1]: 82.92\%\) | \([pf_1]: 87.80\%\) |
| SpfsTu | \([pf_1]: 85.36\%\) | \([pf_1]: 90.24\%\) | \([pf_1]: 95.12\%\) |
| SpfsTu | \([pf_1]: 80.48\%\) | \([pf_1]: 82.92\%\) | \([pf_1]: 90.24\%\) |
| SpfsTu | \([pf_1]: 85.36\%\) | \([pf_1]: 85.36\%\) | \([pf_1]: 92.68\%\) |

Symbol: \(p_f\): point feature; Tu: transient univariate; \(p_{OPT}\): sorted \(p_f\)s of Tu; i: \(p_f\) of Tu j

### TABLE VII

RESULTS OF WRAPPER PHASE BASED ON IWSSR-TWSVM REGARDING DIFFERENT KERNELS PER TRANSIENT UNIVARIATE

| Input | \(p_{OPT}^{IWSS-TWSYMREF}\) | \(p_{OPT}^{IWSS-TWSYMGRBF}\) | \(p_{OPT}^{IWSS-TWSYMPREDK}\) |
|-------|-----------------|-----------------|-----------------|
| SpfsTu | \([pf_1]: 78.04\%\) | \([pf_1]: 80.48\%\) | \([pf_1]: 68.29\%\) |
| SpfsTu | \([pf_1]: 73.17\%\) | \([pf_1]: 85.36\%\) | \([pf_1]: 90.24\%\) |
| SpfsTu | \([pf_1]: 80.48\%\) | \([pf_1]: 82.92\%\) | \([pf_1]: 90.24\%\) |
| SpfsTu | \([pf_1]: 73.17\%\) | \([pf_1]: 85.36\%\) | \([pf_1]: 80.48\%\) |
| SpfsTu | \([pf_1]: 70.73\%\) | \([pf_1]: 90.24\%\) | \([pf_1]: 90.24\%\) |
| SpfsTu | \([pf_1]: 85.36\%\) | \([pf_1]: 87.80\%\) | \([pf_1]: 92.68\%\) |
| SpfsTu | \([pf_1]: 85.36\%\) | \([pf_1]: 85.36\%\) | \([pf_1]: 85.36\%\) |
| SpfsTu | \([pf_1]: 85.36\%\) | \([pf_1]: 85.36\%\) | \([pf_1]: 85.36\%\) |
| SpfsTu | \([pf_1]: 85.36\%\) | \([pf_1]: 85.36\%\) | \([pf_1]: 85.36\%\) |

Symbol: \(p_f\): point feature; Tu: transient univariate; \(p_{OPT}\): sorted \(p_f\)s of Tu; i: \(p_f\) of Tu j
### TABLE VIII
THE OBTAINED FINAL OTFS PER UNIVARIATE (\(f_{OTFs}^{1,10}\)) AND UNION OF OTFS\(^{1,10}\) \(\cup\) \(f_{OTFs}^{1,19}\) BASED ON CROSS-PERMUTATION PHASE OF CQFHS\(S\)

| Input | Permutation: LP | Permutation: RP | Cross | \(f_{OTFs}^{TU}\) |
|-------|-----------------|-----------------|--------|-----------------|
| LLP\(1\) U RLP\(1\)= {\(\{pf_1\}\)} | LLP\(1\) U RLP\(1\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal black sides= {\(\{pf_1\}\)} |
| LLP\(2\) U RLP\(2\)= {\(\{pf_1, pf_6, pf_9\}\)} | LLP\(2\) U RLP\(2\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal dark blue sides= {\(\{pf_1, pf_6, pf_9\}\)} |
| LLP\(3\) U RLP\(3\)= {\(\{pf_1, pf_6, pf_9\}\)} | LLP\(3\) U RLP\(3\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal brown side= {\(\{pf_1, pf_6, pf_9\}\)} |
| LLP\(4\) U RLP\(4\)= {\(\{pf_1\}\)} | LLP\(4\) U RLP\(4\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal light green side= {\(\{pf_1\}\)} |
| LLP\(5\) U RLP\(5\)= {\(\{pf_1, pf_6, pf_9\}\)} | LLP\(5\) U RLP\(5\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal pink side= {\(\{pf_1, pf_6, pf_9\}\)} |
| LLP\(6\) U RLP\(6\)= {\(\{pf_1, pf_6, pf_9\}\)} | LLP\(6\) U RLP\(6\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal red side= {\(\{pf_1, pf_6, pf_9\}\)} |
| LLP\(7\) U RLP\(7\)= {\(\{pf_1, pf_6, pf_9\}\)} | LLP\(7\) U RLP\(7\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal light blue side= {\(\{pf_1\}\)} |
| LLP\(8\) U RLP\(8\)= {\(\{pf_1, pf_6, pf_9\}\)} | LLP\(8\) U RLP\(8\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal green side= {\(\{pf_1, pf_6, pf_9\}\)} |
| LLP\(9\) U RLP\(9\)= {\(\{pf_1, pf_6, pf_9\}\)} | LLP\(9\) U RLP\(9\)= {\(\{pf_1, pf_6, pf_9\}\)} | polygonal yellow side= {\(\{pf_1, pf_6, pf_9\}\)} |

**Symbol:** \(pf\) point feature; \(TU\) transient univariate; \(f_{OTFs}^{TU}\): survived preliminary optimal transient features of \(TU\) by wrapper phase; \(pf_i\) \(i^\text{th}\) of \(TU\); \(U\) union

### C. TSP BASED ON UNION OF OTFS\(^{TU1:TU28}\) (U/OTFS\(^{TU1:TU28}\) SET)

By regarding the above-mentioned points in TTP for TSP, the value of triple indices in TSP (Acc, TPR, and TNR) per fold is shown in Table 10. Based on setting the different value for learning parameters, the fluctuations of the Acc index per fold is recorded, and these values are inserted into Table 10. For example, the Acc variation of fold\(2\) fold\(3\), and fold\(4\) are depicted in Fig. 10. Furthermore, the Table 10 is accompanied by TPR and TNR indices, which is commensurate with the maximum value of Acc in each fold. After extracting all the results based on the evaluation indices, as the overall result of employing the U/OTFS\(^{TU1:TU28}\) in \(SVM^{GRBF}\), calculating the

### TABLE IX
THE EVALUATION METRICS

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

### TABLE X
RESULTS OF TSP BASED ON U/OTFS\(^{TU1:TU28}\) SET

| Classifier | Test case | 10-fold cross validation | Max(Acc) per fold based on fine-tuning on \(C\) and \(\sigma\) | Accuracy [TPR / TNR] |
|------------|-----------|-------------------------|------------------------|---------------------|
| SVM\(^{GRBF}\) NETS-NYPS | | | | |
| 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] |
| 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] | 98.87 | [98.5 / 99.25] |
average of the Acc, TPR, and TNR is regarded in Table 10. The results of Table 10 show the fact that the $U_f^{\text{OTFs}}_{TU1:TU28}$ set plays the pivot role in raising the capacity of prediction (high DLA) in TSA (Acc: 98.87 %, TPR: 98.5 %, and TNR: 99.25 %). Besides the importance of DLA, we concentrated on the DLT factor (including observed time and prediction time) in the performance evaluation of proposed framework in TSA. For DLT calculation, we need to focus on selected point features $U_f^{\text{OTFs}}_{TU1:TU28}$ (See Table 8). The optimal features in the $U_f^{\text{OTFs}}_{TU1:TU28}$ set show the fact the selected cycles of some univariates are related to the 9th cycle. (e.g., $f^{\text{OTFs}}_{TU15}$, $f^{\text{OTFs}}_{TU17}$, $f^{\text{OTFs}}_{TU18}$, and $f^{\text{OTFs}}_{TU25}$). Hence, observed time (OT) is equal to 9 cycles (150.3 milliseconds (ms)), and the prediction time based on $SVM^{GRBF}$ - $U_f^{\text{OTFs}}_{TU1:TU28}$ is 2.225 ms. Consequently, the DLT is 152.525 ms (See Table 11), indicating a low DLT to address corrective control actions.

### Table XI

| Observed Time (OT) | Data Labeling Time (DLT) |
|--------------------|--------------------------|
| $9 / 0.1503$      | $150.3 ms + 2.225 ms = 152.525 ms$ |

D. Comparison of Experimental Methods: CPQHFSS Vs. Vertically Unilateral FSSs (VUFSSs) and MHFSS

For further evaluation on the efficacy of CPQHFSS-based $U_f^{\text{OTFs}}_{TU1:TU28}$ in TSP, the comparison between with three VUFSSs (3VUFSSs) is considered in this section. The 3VUFSSs includes mRMR [16, 17], ReliefF [18] and fast correlation-based filter (FCBF) [19]. Also, we compared the CPQHFSS with the bi-mode hybrid feature selection scheme (BMHFSS) [23] and partial-injective trilateral hybrid (filter-wrapper) scheme called PITHS [34] as the MHFSS. By applying the 28-variate time-series data to 3VUFSSs and 2MHFSS, the 3VUFSSs-based OTFs and 2MHFSS-based OTFs are extracted. Next, the 3VUFSSs and 2MHFSS are entered into the

![FIGURE 10. Acc variations based on learning parameters in some folds (fold 2, 5, 7, 10) for TSP based on $U_f^{\text{OTFs}}_{TU1:TU28}$](image-url)
SVM\(^{GRBF}\) learning model in the same learning conditions expressed in subsection \(C\) of section 3 (10-fold cross-validation technique and the fine-tuning of the learning parameters).

As can be seen in Table 12, CPQHFSS\(^{OTFs}\) have better performance in TSP than 3VUFSS\(^{OTFs}\) and 2MHFSS\(^{OTFs}\) (ignoring only 0.25% less than TPR than PITHS). The obtained results of Table 12 show that CPQHFSS in the

| TABLE XII |
|---|
| RESULTS OF TSP BASED ON SELECTED OTFS BY 3VUFSS AND 2MHFSS |

| (FSS) Classifier | Test case | 10-fold cross validation | Mean(measure) of folds: Accuracy [TPR / TNR] |
|---|---|---|---|
| mRMR | NETS-NYPS | SVM\(^{GRBF}\) | | |
| 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] |
| 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] |
| 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] |
| 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] |
| fold 1 | fold 2 | fold 3 | fold 4 |
| 97.5 | 95 | 96.25 | 97.5 |
| [95 / 100] | [95 / 95] | [92.5 / 100] | [95 / 100] |
| 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] | 98.75 [97.75 / 98.75] |
presence of 48-cycles of 28-variate trajectory (See Table 8) has better performance (Acc, TPR, and TNR) than mRMR (9-cycle of 4-variate trajectory features), FCBF, ReliefF, and BMHFSS (9-cycle of 3-variate trajectory features [23]. In terms of the PITHS, method, the selected cycles via CPQHFFSS leading to high Acc and TPR than PITHS (24-cycle of 18-variate trajectory features [34]). In terms of DLT, CPQHFFSS has higher DLT (152.525 ms) than 3VUFSSs (mRMR: 68.793 ms, FCBF: 68.930 ms, ReliefF: 68.910 ms) MHFSS\textsuperscript{BMHFF} (BMHFSS: 52.948 ms). Also, based on DLT index, CPQHFFSS has lower DLT than MHFSS\textsuperscript{PITHS} (PITHS: 152.591 ms) (See Table 13). Generally, the CPQHFFSS\textsuperscript{DLT} causes the system operator to have enough time to take corrective actions. The detail information about CPQHFFSS\textsuperscript{DLT}, 3VUFSSs\textsuperscript{DLT}, and 2MHFSS\textsuperscript{DLT} are shown in Table 11 and Table 13.

TABLE XIII

| 3VUFSSs \& 2MHFSS | Observed Time (OT in cycle / second) | DLT (OT + prediction time) |
|-------------------|-------------------------------------|-----------------------------|
| mRMR              | 4 / 0.0668                           | 66.8 ms + 1.993 ms = 68.793 ms |
| FCFB              | 4 / 0.0668                           | 66.8 ms + 2.130 ms = 68.930 ms |
| ReliefF           | 4 / 0.0668                           | 66.8 ms + 2.110 ms = 68.910 ms |
| BMHFSS            | 3 / 0.0501                           | 50.1 ms + 2.848 ms = 52.948 ms |
| PITHS             | 9 / 0.1503                           | 150.3 ms + 2.291 ms = 152.591 ms |

V. CONCLUSION

The presence of tightly correlated indices in the TSHA problem, namely, accuracy and time of transient prediction, extracting optimal transient features (OTFs) from HDTF space via multifaceted FSS to meet low DLT and high DLA, was defined as the main agenda of this paper. To this end, we proposed cross-permutation-based quad-hybrid FSS (CPQHFFSS) designed by integrating four filter-wrapper blocks (FWBs) in the form of twin-two-FWBs mounted on dual incremental wrapper mechanisms (IWMs) called IWSS\textsuperscript{F2WBs} and IWSS\textsuperscript{S2PBs}. Regarding filter-fixed and wrapper-varied approaches (FWiv) embedded into IWMS\textsuperscript{S2PBs} of CPQHFFSS, the relevancy ratio-support vector machine (RR-SVM) raised as FW in the first block of IWSS\textsuperscript{F2WBs} and IWSS\textsuperscript{S2PBs}, and relevancy ratio-twin support vector machine (RR-TWSSVM) situated in the second block of IWSS\textsuperscript{F2WBs} and IWSS\textsuperscript{S2PBs}. Furthermore, the nonlinear nature of transient feature space makes the plugging elastic and non-elastic kernels into the wrapper phase an integral part of CPQHFFSS. Hence, by selecting the triple nonlinear kernel, cross-permutations of hybrid FSS are applied on HDTF to select OTFs. After applying CPQHFFSS on transient multivariate trajectories, selected optimal features are fed to the cross-validation scenario to evaluate their effectiveness on TSHA. Experimental results show that the selected OTFs based on the CPQHFFSS have a high performance (Acc 98.87%, TPR 98.5%, TNR 99.25%, and DLT of 152.525 ms) for TSP. For more clarity, the performance of CPQHFFSS compared with 3VUFSSs and 2MHFSS. The obtained results show that the selected CPQHFFSS-based OTFs have better performance than selected OTFs by mRMR, ReliefF, FCBF, BMHFSS, and PITHS algorithms on TSP.

REFERENCES

[1] S. Gao, S. Qiu, Z. Ma, R. Tian, and Y. Liu, “SVAE-WGAN-Based Soft Sensor Data Supplement Method for Process Industry,” IEEE Sensors Journal, vol. 22, no. 1, pp. 601-610, Jan. 2022.
[2] Z. You and L. Feng, “Integration of Industry 4.0 Related Technologies in Construction Industry: A Framework of Cyber-Physical System,” IEEE Access, vol. 8, pp. 122908-122922, July 2020.
[3] M. I. Alabsi and A. Q. Gill, “A Review of Passenger Digital Information Privacy Concerns in Smart Airports,” IEEE Access, vol. 9, pp. 33769-33781, Feb. 2021.
[4] L. Nardi, O. Pinheiro, and H. Lepikson, “Computer System Integrated with Digital Models for Reconstruction of Underwater Structures with High Definition,” IEEE Latin America Transactions, vol. 20, no. 2, pp. 283-290, Feb. 2022.
[5] M. A. A. Sufyan, M. Zuhair, M. A. Anees, A. Khair, and M. Rihan, “Implementation of PMU-Based Distributed Wide Area Monitoring in Smart Grid,” IEEE Access, vol. 9, pp. 140768-140778, Oct. 2021.
[6] J. Han, M. Kamber, and J. Peri, J. K, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
[7] 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.
[8] G. V. Trunk, “A Problem of Dimensionality: A Simple Example,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, no. 3, pp. 306-307, July. 1979.
[9] K. Kourtoumbas and S. Theodoridis, Pattern Recognition, 1st ed. Academic Press: Elsevier, 2008.
[10] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning, 1st ed. New York, NY, USA: Springer, 2013.
[11] V. Bolón-Canedo, N. Sánchez-Marono, and A. Alonso-Betanzos, Feature Selection for High-Dimensional Data, 1st ed. Springer Cham, 2016.
[12] S. Das, S. P. Singh, and B. K. Panigrahi, “Transmission line fault detection and location using wide area measurements,” Electric power Syst. Res., vol. 151, pp. 96–105, Oct. 2017.
[13] 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.
[14] C.-S.-G. Karavas, K. A. Plakas, K. F. Kromydas, A. S. Kurashvili, C. N. Dikaiakos, and G. P. Papanoamou, “A review of wide-area monitoring and damping control systems in Europe,” in Proc. IEEE Madrid PowerTech, Madrid, Spain, Jun. 2021, pp. 1–6.
[15] M. Pavelia, 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.
[16] 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.
[17] 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.
[18] A. Stief, J. R. Otewill, and J. Baranowski, “Relief F-based feature ranking and feature selection for monitoring induction motors,” in Proc. 25th Int. Conf. Methods Models Autom. Robot. (MMAR), Miedzyzdroje, Poland, Aug. 2018, pp. 171–176.
[19] 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, Nov. 2018, pp. 952–957.
[20] S. Liu, L. Liu, Y. Fan, L. Zhang, Y. Huang, T. Zhang, J. Cheng, L. Wang, M. Zhang, R. Shi, and D. Mao, “An integrated scheme for online dynamic security assessment based on partial mutual information and
iterated random forest,” IEEE Trans. Smart Grid, vol. 11, no. 4, pp. 3606–3619, Jul. 2020.
[21] 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.
[22] 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.
[23] 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.
[24] 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, May 2009, pp. 367–374.
[25] D. Tomar and S. Agarwal, “Twin Support Vector Machine,” Egyptian Informatics Journal, vol. 16, no. 1, pp. 55-69, Mar. 2015.
[26] 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.
[27] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Burlington, MA, USA: Morgan Kaufmann, 2011.
[28] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, pp. 273–297, Apr. 1995.
[29] O. L. Mangasarian and E. W. Wild, “Multisurface Proximal Support Vector Classification via Generalized Eigenvalues,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pp. 69-74, Jan. 2006.
[30] 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.
[31] C. M. Bishop, Pattern Recognition and Machine Learning, 1st ed. NY: Springer, 2006.
[32] H. Shimoda, 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, 2001, pp. 1–4.
[33] P.-F. Marteau and S. Gibet, “On recursive edit distance kernels with application to time series classification,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 6, pp. 1121–1133, Jun. 2015.
[34] 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, Dec. 2021.
[35] 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.
[36] 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.
[37] S. A. B. Mosavi, A. Amiri, and S. 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, Nov. 2018.
[38] 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.
[39] S. A. B. Mosavi and S. Banhashem, “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.