energies

MDPI

Article

Mitigating Missing Rate and Early Cyberattack Discrimination Using Optimal Statistical Approach with Machine Learning Techniques in a Smart Grid

Nakkeeran Murugesan [1], Anantha Narayanan Velu [1,*], Bagavathi Sivakumar Palaniappan [1], Balamurugan Sukumar [2] and Md. Jahangir Hossain [3]

[1] Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore 641112, India; m_nakkeeran@cb.students.amrita.edu (N.M.); pbsk@cb.amrita.edu (B.S.P.)
[2] Department of Electrical and Electronics Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore 641112, India; s_balamurugan@cb.amrita.edu
[3] School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia; jahangir.hossain@uts.edu.au
* Correspondence: v_ananthanarayanan@cb.amrita.edu

Abstract: In the Industry 4.0 era of smart grids, the real-world problem of blackouts and cascading failures due to cyberattacks is a significant concern and highly challenging because the existing Intrusion Detection System (IDS) falls behind in handling missing rates, response times, and detection accuracy. Addressing this problem with an early attack detection mechanism with a reduced missing rate and decreased response time is critical. The development of an Intelligent IDS is vital to the mission-critical infrastructure of a smart grid to prevent physical sabotage and processing downtime. This paper aims to develop a robust Anomaly-based IDS using a statistical approach with a machine learning classifier to discriminate cyberattacks from natural faults and man-made events to avoid blackouts and cascading failures. The novel mechanism of a statistical approach with a machine learning (SAML) classifier based on Neighborhood Component Analysis, ExtraTrees, and AdaBoost for feature extraction, bagging, and boosting, respectively, is proposed with optimal hyperparameter tuning for the early discrimination of cyberattacks from natural faults and man-made events. The proposed model is tested using the publicly available Industrial Control Systems Cyber Attack Power System (Triple Class) dataset with a three-bus/two-line transmission system from Mississippi State University and Oak Ridge National Laboratory. Furthermore, the proposed model is evaluated for scalability and generalization using the publicly accessible IEEE 14-bus and 57-bus system datasets of False Data Injection (FDI) attacks. The test results achieved higher detection accuracy, lower missing rates, decreased false alarm rates, and reduced response time compared to the existing approaches.

Keywords: blackouts; cascading failures; cyberattacks; feature extraction; intrusion detection system; machine learning; smart grid

Citation: Murugesan, N.; Velu, A.N.; Palaniappan, B.S.; Sukumar, B.; Hossain, M.J. Mitigating Missing Rate and Early Cyberattack Discrimination Using Optimal Statistical Approach with Machine Learning Techniques in a Smart Grid. *Energies* **2024**, *17*, 1965. https://doi.org/10.3390/en17081965

Academic Editors: José Matas, Jorge El Mariachet and Sen Tan

Received: 21 March 2024
Revised: 12 April 2024
Accepted: 15 April 2024
Published: 20 April 2024

1. Introduction

The mission-critical infrastructure of Cyber–Physical Power Systems [1] (CPPS), such as a smart grid, has been targeted for cyberwarfare to cause physical sabotage, large-scale load loss, blackouts, and cascading failures [2–4]. In most cases of power grid disturbances, fault analysis and diagnosis [5] are conducted using state estimation methods [6] and time-series analysis [7]. However, the continuously increasing rate of network traffic makes it difficult for cyber analysts to spot new patterns of behavior in the network. The current data volume, velocity, and variety across firewalls make it more difficult for cyber analysts to monitor successfully. Moreover, 61% of firms admit that, without Artificial Intelligence (AI), they cannot spot critical threats [8]. In the current era of Industry 4.0, researchers and

scientists have started to apply AI in cybersecurity to employ Intelligent IDS in Wide Area Measurement Systems (WAMS) to protect smart grids from advanced cyberattacks [9,10]. These challenges have motivated researchers to provide a defensive in-depth approach to predict anomalies earlier for smart grids with machine learning techniques [11–14] and deep learning techniques [15,16]. Moreover, to enhance the cybersecurity aspects in smart grids, a study on False Data Injection (FDI) attacks provides insights into the paradigm shift in power systems to digitalization, the vulnerability of the protocols, various detection methods, and a mitigation strategy [17].

Generally, an IDS is broadly classified into one of three distinct groups: Signature-based IDS, Specification-based IDS, or Anomaly-based IDS [18,19]. A Signature-based IDS will not be sufficient for the growing cyber threats from motivated attackers since it often needs to be updated. It is limited to known attacks and fails to recognize unknown attacks. A Specification-based IDS is complex and resource-intensive, and it requires system expertise, which is partially effective in identifying unknown attacks [20]. In comparison, an Anomaly-based IDS can detect and recognize unknown attacks but with a high FPR. So, a robust Anomaly-based IDS is required to detect anomalies with higher accuracy, low false negatives (missing rate), and low false positives (false alarms) with decreased response times. In the mission-critical infrastructure of a smart grid, even a few instances of misclassification might have fatal consequences regarding the power system's stability and reliability, which necessitates thorough investigation [21]. Our proposed SAML-Triple approach is an alternative solution to the existing Specification-based IDS (state estimation approach) and Signature-based IDS.

The scope of this paper includes developing a robust Anomaly-based IDS that discriminates cyberattacks from natural faults and man-made events with an early attack detection mechanism with reduced missing attacks, increased specificity, and fewer false alarms with high detection accuracy. This work will support the nation's smart grid mission and private industries to provide in-depth defense in detecting cyberattacks at the physical layer when the system is in a critical situation compromised by an attacker or intruder within the system or externally. The significant contributions of this paper are presented below:

- The novel mechanism of a **S**tatistical **A**pproach with a **M**achine **L**earning (**SAML**) classifier based on Neighborhood Component Analysis (NCA), ExtraTrees, and AdaBoost for feature extraction, bagging, and boosting, respectively, is proposed with optimal hyperparameter tuning for the early discrimination of cyberattacks in a smart grid with the three-bus/two-line transmission system of triple-class datasets (No Events/Natural Events/Attack Events);
- The proposed model is evaluated for generalization and scalability with IEEE 14-bus and 57-bus system datasets of False Data Injection (FDI) attacks to prove the robustness;
- The missing rate is handled in this paper through **INF**inity **A**ttack Records as **Z**ero (**INFAZ**) to avoid blackouts and cascading failures and INFinity **A**ttack Records by **D**ropping (**INFAD**) to improve the accuracy;
- The early response time is set to less than 8.3 ms on 120 samples/a second system to detect the attack before the system collapses.

The proposed SAML-Triple approach considers the preprocessing aspects of both INFAZ and INFAD. The two distinct aspects of comparison are aimed at overcoming the existing drawbacks, as discussed in the related work in Section 2. SMOTE is applied to balance the dataset by considering an equal number of samples from each class with stratified sampling. The train–test split of an 80:20 ratio is taken for the SAML-Triple approach.

This paper is split into seven sections: Section 2 discusses the related work regarding triple-class classification with the proposed techniques, drawbacks of the existing approaches, and challenges addressed. Section 3 describes the proposed approach with a process flow diagram of the preprocessing techniques of feature engineering: handling "INFinity" Attack Events records with two preprocessing aspects (INFAZ and INFAD), feature scaling, handling imbalanced data using SMOTE, and the statistical approach of

NCA with a hyperparameter optimization strategy combined with a machine learning classifier. Moreover, the description of the publicly available ICS Cyber Attack Power System datasets with operational scenario categories is presented in detail. Section 4 describes the proposed methodology, which deals with the statistical approach of feature extraction techniques using NCA and optimal parameter/hyperparameterized tuning with the (ET + AdB) ML classifier. Algorithm 1 represents the Pseudocode for Data Preprocessing and NCA transformation, whereas Algorithm 2 represents the Pseudocode for Optimal Hyperparameter tuning to find the 'N' *Component*, and Max. *Iteration* 'I' of NCA with the best parameters for each ML classifier is applied. Section 5 provides the implementation details of the data preparation, hyperparameter settings for the models, test case scenarios, tools for implementation, and evaluation metrics. Section 6 includes the results analysis and discussion of the three-bus/two-line transmission system (triple-class dataset) and IEEE 14- and 57-bus systems datasets of FDI attacks for generalization and scalability in detail using tables, graphs, and a confusion matrix. The performance metrics of FNR (or missing rate), response time, FPR (or false alarm), and accuracy are considered to compare the results. Finally, Section 7 discusses the conclusion and the scope of future work.

2. Related Works

In the related works, some researchers recently addressed the problem of discriminating cyberattacks from Natural Events and No Events with their proposed techniques and approaches.

Upadhyay et al. [22] used Gradient Boosting Feature Selection (GBFS), an ensemble learning technique, to reduce the features from 128 to 15 features and obtained an accuracy of 96.50% with a tree-based machine learning classifier. GBFS combines multiple weak learners' predictions to create a strong predictive model. The removal of the features based on feature importance scores provided by GBFS may not necessarily imply the improvement of the model. Some features might be important in combination with others, and removing them individually might not result in a better-performing model. GBFS relies solely on feature importance scores, which limits the feature selection in the triple-class dataset. Also, each of the 15 datasets yields a different combination of 15 features from the 128 features in Table 1. The authors were not convinced as to which 15 features were the best among the 15 datasets to discriminate Attack Events from Natural Events and No Events. Furthermore, the same author group, Upadhyay et al. [23], proposed an integrated framework for an IDS for SCADA-based power grids, in which they used Recursive Feature Elimination (Feature Selection Technique) to reduce the features from 128 to 30 features and nine heterogeneous ensemble classifiers with the majority voting stacking concept to achieve improved accuracy of 97.95% for the same triple-class dataset. Recursive Feature Selection (RFE) removes the least important features based on a model's performance until the desired number of features is selected. The drawback of the RFE approach is that it may struggle when dealing with highly correlated features, and removing one of a set of correlated features might not necessarily improve the model's performance. It could lead to the loss of valuable information, with degraded performance in extracting the features and discriminating the Attack Events from Natural Events and No Events. The **first drawback** of [22,23] is that the authors' perspectives on preprocessing the "INFinity" Attack Events records are contradictory. Moreover, the **second drawback** is that, in [22], the author group mentioned the top 15 promising influenced features for attack classification; in contrast, in [23], they did not list the top 30 influencing features, which contradicts the statement of promising features from 15 to 30. However, they achieved a slight improvement in accuracy of 97.95% in [23] compared to [22], with an accuracy of 96.50% for the triple-class datasets.

Hu et al. [24] used a Stacked Denoising Autoencoder (SDAE) to extract the features from 128 to 60 features. They classified them using an Extreme Gradient Boosting (XGBoost) classifier for the triple-class dataset with an accuracy of 90.48%. The authors used the deep learning model of SDAE to learn the complex input data representation to extract the features. SDAEs have several hyperparameters, including the number of layers, the

number of nodes in each layer, learning rates, and corruption levels. Selecting appropriate hyperparameters is challenging, and suboptimal choices result in the model's poor performance in extracting the features and discriminating the Attack Events from Natural Events and No Events. Moreover, the same author group, Hu et al. [25], used Multiple Autoencoders (AE) to extract the features from 128 to 30 features and classified them using a Random Forest machine learning classifier to discriminate the triple-class dataset, with an improved accuracy of 91.78%. Multiple Autoencoders (AE) learn the complex representation of input data to extract features. Hyperparameter tuning of Multiple Autoencoders and the adaptive boosting mechanism is challenging. It requires extensive experimentation to find an optimal combination. This results in the degradation of model performance in extracting the features and discriminating the Attack Events from Natural Events and No Events. The **first common drawback** of both works [24,25] is that they have not mentioned the preprocessing of "INFinity" Attack Events records. The **second drawback** is that the feature selection is inconclusive, with wide variation in selecting optimal features, as they mentioned 60 features in [24] and 30 features in [25], a slight improvement in accuracy from 90.40% to 91.78%.

Gumaei et al. [26] used the Correlation-based Feature Selection Technique for selecting the optimal features with KNN as a machine learning classifier to discriminate the triple-class datasets with an accuracy of around 91.87%, where each of the 15 datasets was processed individually to select the optimal features, with variation regarding eight to eleven features. Correlation-based feature selection measures linear relationships between variables, and it may not capture the true association between the features and the target variable if the relationships are non-linear. Also, it may not capture complex interactions or dependencies involving multiple features simultaneously. It results in a poor choice of feature selection, which may not be accurate enough to discriminate Attack Events from Natural Events and No Events. Moreover, this paper **recommended** the future scope of increasing the accuracy with less computational time based on hybrid feature selection techniques.

Ankitdeshpandey and Karthi, R. [27] applied Principal Component Analysis (PCA) as an unsupervised feature extraction technique to reduce the dimensionality to 31 principal components to discriminate the triple-class dataset using ML and DL classifiers with an accuracy of 91.14% for Random Forest, 89.91% for DNN, and 76.90% for SVM, where all three of the classifier results demonstrated meager detection rates. PCA focuses on capturing the global variance in the data along the principal components in which the data vary. It does not preserve the local structure or relationships within the data, which might result in misclassification between the three classes. The **limitation** of this paper is that they tested only for the reduced amount of around 13,200 samples with random selection from the entire 15 datasets.

Hink et al. [28] developed the **original datasets** by considering the scenario of an insider attack (or compromised system) in a smart grid. They investigated power system disturbances and cyberattack discrimination using machine learning applications. This author's group's dataset provides the first proof for carrying out research in machine learning applications to develop an IDS. The **limitation** of this paper is that they tested only for 1% of the randomly sampled records from the entire 15 datasets across all three classification formats: Binary Class, Triple Class, and Multiclass. The sample measurement considered 294 records of "No Events", 3711 Attack Events, and 1221 Natural Events records used across all three classification formats. Using the Information Gain as a Feature Selection Technique, 40 features were selected as optimal features and discriminated against the triple-class dataset using the Adaboost + JRipper ML classifier with an accuracy of 95.0%. Information Gain assumes that features are independent, and this assumption is often violated in the case of highly correlated triple-class datasets. As a result, Information Gain may not accurately reflect the true importance of features, leading to suboptimal feature selection, which may not be efficient in discriminating between the three classes. This original author group recommended evaluating the future scope of work with a broader

range of power system data, learning algorithms, classification strategies, and labeled data amounts.

Agrawal et al. [29] applied the concept of dynamic retraining with drift detection toward robust power grid attack protection using LightGBM. They classified the triple-class dataset with an accuracy of 95.30% for the complete 128 features and 97.1% for the top selected ten features using the ExtraTrees approach as the feature selection technique. The **drawback** of this paper is that they removed the "INFinity" Attack Events records, which may lead to missing rates or SCADA inoperability. The feature importance scores provided by ExtraTrees as a feature selection method may not always reflect the true importance of features, especially in the presence of highly correlated features. The algorithm may arbitrarily assign importance to one of the correlated features, leading to potential bias in feature selection and misclassification between the three classes.

Sunku Mohan et al. [30] investigated the problem using Power Domain Knowledge. These authors employed manual feature selection by filtering out features of positive-, negative-, and zero-sequence components and logs. They selected 36 features manually for this triple-class dataset for discriminating cyberattacks, Natural Events, and load variation in SCADA smart grid systems. They applied a Rule-based Machine Learning Adaboost classifier to discriminate the triple-class dataset with an accuracy of 97.25%. The **limitation** of this paper is that manual feature selection is more domain-specific, which is a partial specification-based IDS, even though the classification was performed through a machine learning classifier. This manual feature selection may not be suitable for the generalizability and scalability of the model for different architectures and may require complex logical calculations.

Bitirgen K and Filik ÜB [31] developed a hybrid model by combining particle swarm optimization (PSO) with convolutional neural network (CNN) and long short-term memory (LSTM), PSO-CNN-LSTM, to optimize the features for better triple-class classification, with an accuracy of 96.92%. The authors utilized PSO as a metaheuristic optimization algorithm for better search space, along with CNN to develop input features with complicated mathematical operations and LSTM to preserve both the short-term and long-term dependencies of time-series data. The **drawback** of this paper is that they did not mention the preprocessing of "INFinity" Attack Events records and the number of feature selections, even though they achieved better detection accuracy. In the case of a highly correlated triple-class dataset, the effectiveness of PSO is influenced by the starting positions and velocities of particles, and it may struggle to handle redundant features effectively. If multiple features are highly correlated, PSO might select only one, leading to potential bias in the feature selection and misclassification between the three classes. The graphical representation shown in this paper is a mere feature representation before applying the model and does not better represent the cluster of the labels.

In the solving approaches, NCA indirectly involves eigenvalues and eigenvectors in the computation of transformation matrix A. In [32], the author efficiently used eigenvalues and eigenvectors for optimal sensor placement with multi-objective robust optimization. In [33], the author incorporated Adaptive PCA (A-PCA) for extracting the best features from the network traffic for the IDS. The eigenvalues and eigenvectors of the covariance matrix provide valuable information regarding the principal axes and magnitudes of variation in the data. This information can guide the selection of appropriate dimensions or components for representation.

2.1. Drawbacks and Challenges of the Existing Approaches

- The drawbacks and challenges faced by most of the researchers are that they tried to discriminate the No Events/Natural Events/Attack Events of triple-class datasets with different **feature selection numbers varying from eight to sixty features** with their adopted feature selection techniques [22,23,26,28,29,31] to improve the detection accuracy. The feature importance scores obtained from the suboptimal feature selection of the existing methods indicate poor choices of features, leading to potential bias in

feature selection and a lack of model performance when discriminating against attacks. The authors [24,25,27] used Autoencoders as dimensionality reduction techniques of **feature extraction** to reduce the features using unsupervised techniques. The feature extraction using deep learning methods lacks the optimal combination of extracting the features due to several hyperparameter factors. Moreover, dimensionality reduction techniques (feature extraction) using PCA [27] fail to capture the local structure or relationships within the data, which might result in misclassification between the three classes.

- The other major drawback is that **dropping** the feature column of "PA: Z" (Apparent Impedance for Four Relays) or **removing** the "INFinity" Attack Events records rows seems to be quite contradictory as those researchers attempted to avoid the attack scenarios, which might lead to increases in the missing rate or false negative rate. If left unprocessed, it may have a massive impact on the SCADA systems, making them inoperable. It might result in fatal consequences regarding the power system's stability and reliability. We have addressed this problem in our proposed SAML-Triple approach by considering it zero (INFAZ).

2.2. Research Gap Identified

Due to the highly complex correlation between the features of the Triple Class Power System Cyberattack Dataset, the existing approaches are ineffective in selecting the optimal subset of features. The existing approaches [22,23,26,28,29,31] do not select the suboptimal features with feature importance scores, leading to **potential feature selection bias**. The existing approaches do not effectively discriminate cyberattacks from Natural Events and No Events. The rest of the existing approaches [24,25,27] use the feature extraction techniques of deep learning methods like Autoencoders and PCA. Such techniques do not result in the optimal combination of extracting the features due to several **hyperparameter factors**. Moreover, the existing approaches provide **less accuracy** in the **discrimination of cyberattacks**. We propose an alternative approach of SAML-Triple to overcome the shortcomings of the existing approaches in discriminating cyberattacks from natural faults and man-made events. **The proposed work focuses** on **improving the accuracy** as well as **mitigating the missing rates** with **earlier cyberattack detection**.

2.3. Addressing the Challenges through the Proposed Approach

- To avoid ambiguity regarding the different number of feature selections, we utilized the NCA as a supervised feature extraction method for dimensionality reduction. This method does not rely on feature importance scores; instead, it converts high-dimensional data into lower-dimensional data in a new transformation space suitable for complex and highly correlated data. It preserves both the global and the local neighborhood structure relationship between the data records in the dataset. The proposed **SAML-Triple** adopts NCA as a feature extraction technique by optimal parameter/hyperparameter tuning with the (ET + AdB) ML classifier in discriminating cyberattacks from natural faults and man-made events.
- We addressed the "INFinity" Attack Events records in the feature column of "PA:Z" (Apparent Impedance for Four Relays) by replacing them with "Zero" (**INFAZ**). In the context of the power domain, the "PA:Z" – INFinity value can be processed either as "Zero" or the range of value above its limit to avoid the missing rate [21]. Here, two preprocessing aspects of INFAZ (Zero) and INFAD (Dropping) in the SAML-Triple work were performed to compare the results with those of other existing works.

3. Proposed Approach

The Process Flow Diagram in Figure 1 stands for the steps involved in the SAML-Triple involving feature engineering aspects and the hyperparameter optimization strategy of the model for discrimination of cyberattacks from natural faults and man-made events.

In **SAML-Triple**, the source of datasets considers the publicly available ICS Cyber Attack Triple Class (No Events/Natural Events/Attack Events) Power System Datasets [34]. In SAML-Triple, the data preprocessing steps in feature engineering are carried out with two aspects: handling "INFinity" Attack Events records as zero (INFAZ) and dropping "INFinity" Attack Events records (INFAD) for the feature columns of "PA:Z" (Apparent Impedance for Four Relays). Adopting two distinct data preprocessing aspects in the SAML-Triple is to compare the results with the existing approaches. In continuation with feature engineering, Standard Scalar, and Label Encoders were applied, followed by SMOTE to balance the dataset, considering the equal number of records from each class label with stratified sampling. For both aspects, optimal features are extracted by utilizing NCA as a feature extraction technique to find the Optimal 'N' *Component* with a Maximum number of *Iterations* 'I'. In SAML-Triple, the train–test split of an 80:20 ratio was applied. After the train–test split process, optimal hyperparameter tuning with GridSearchCV (10-fold cross-validation) for each of the ML classifiers is applied to exhaustively search for the best parameters from the grid of provided parameters.

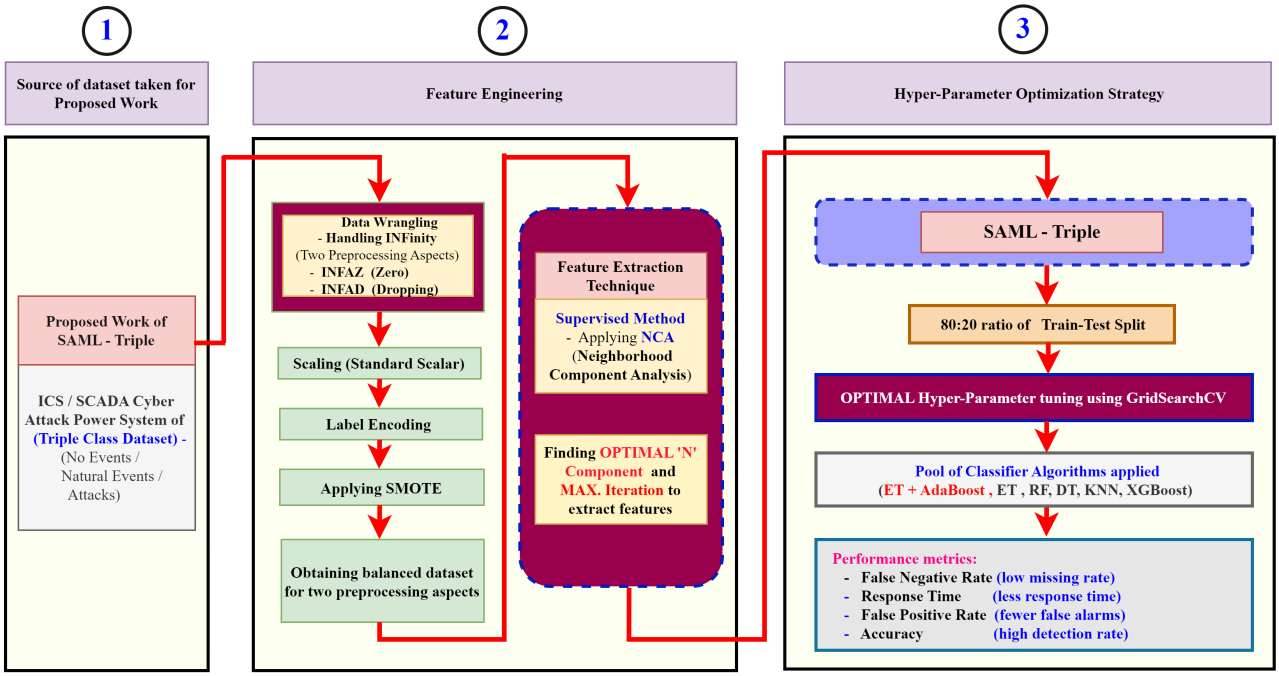


Figure 1. Process flow diagram of the SAML-Triple approach of Anomaly-based IDS in a smart grid using dataset [34].

A pool of ML classifier algorithms [35] is applied for training and testing the datasets with ExtraTrees with AdaBoost classifier (ET + AdB), ExtraTrees (ET), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN), and XGBoost (XGB). The **SAML-Triple** approach of NCA with the (ET + AdB) ML classifier performs well in discriminating cyberattacks on both aspects with better performance metrics. The performance metrics of FNR (or missing rate), Response time, FPR (or false alarm), and Accuracy were compared in this work with various ML classifiers.

3.1. Industrial Control Systems (ICS) Cyber Attack Power System Dataset Testbed Description

The publicly available ICS Cyber Attack Power System Dataset [34] was developed by Mississippi State University in collaboration with Oak Ridge National Laboratory in 2014.

Figure 2 represents the power system framework configuration of a 3-Bus/2-generator system developed by the authors [36]. The assumptions are made that an intruder has already compromised the system, acquired access to the substation network, and sent commands to the substation switch. The intruder (or attacker) could be a former employee of the company or

present employee or an external source from outside the network. Since the IEDs (Intelligent Electronic Devices) lack an internal validation system to distinguish between genuine and fraudulent faults, they employ a distance protection technique to trip the breakers on detected faults. Operators can manually trip breakers BR1 through BR4 by sending orders to IEDs R1 through R4. Usually, manual overriding is performed during line maintenance or when other system components fail. SAML-Triple considers this framework, which comprises various operational scenarios [34], such as *Single Line-to-Ground, line maintenance, remote tripping command injection attack, relay setting change attack, and FDI attack* to ensure that cyberattack discrimination is valid during normal routine operations, including manipulated breakers.

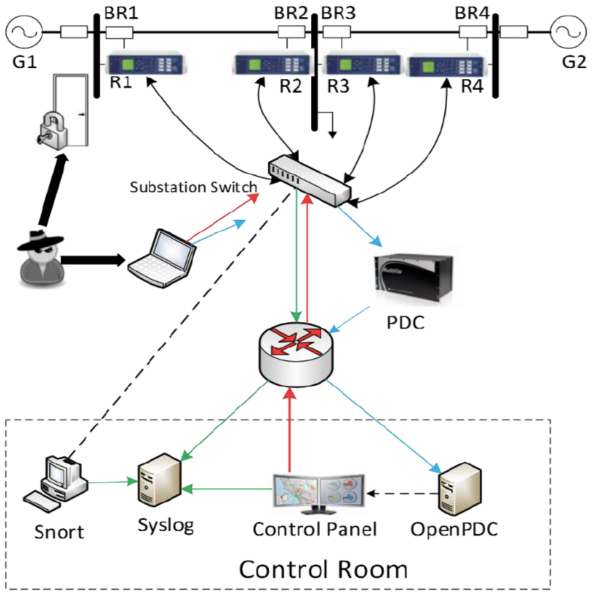


Figure 2. The power system framework configuration (3-bus/2-line transmission system) [34].

The SAML-Triple considers 15 triple-class datasets for evaluation to discriminate cyberattacks from Natural Events and No Events. Each of the 15 datasets has an approximately equal number of 5000 records with 128 feature columns and one marker column as the target label for classification. Table 1 provides the dataset's feature description of 128 features. A detailed description of the features dataset is available in [34]. Various types of operation scenarios (37 power system events) are categorized into three labels (No Events (1)/Natural Events (8)/Attack Events (28)). Table 2 shows the SAML-Triple class with detailed 41 event scenario splits into three class labels, namely—No Events (41), Natural Events (1 to 6, 13, 14), and Attack Events (7 to 12, 15 to 20, 21 to 30, and 35 to 40).

- **No Events**—stands for the normal system operation with changes in loads.
- **Natural Events**—stands for the system with Single Line-to-Ground (SLG) with various percentages of fault location in L1 and L2 with the addition of Line Maintenance (L1 and L2).
- **Attack Events**—stands for the data injection attack (SLG fault replay), remote tripping command injection attack, and relay setting change attack with various percentages of fault location.

Table 1. Dataset feature descriptions [34].

Feature	Description
PA1:VH – PA3:VH	Phase A – C Voltage Phase Angle
PM1:V – PM3:V	Phase A – C Voltage Phase Magnitude

Table 1. *Cont.*

Feature	Description
PA4:IH – PA6:IH	Phase A – C Current Phase Angle
PM4:I – PM6:I	Phase A – C Current Phase Magnitude
PA7:VH – PA9:VH	Pos. – Neg. – Zero Voltage Phase Angle
PM7:V – PM9 :V	Pos. – Neg. – Zero Voltage Phase Magnitude
PA10:IH – PA12:IH	Pos. – Neg. – Zero Current Phase Agle
PM10:I – PM12:V	Pos. – Neg. – Zero Current Phase Magnitude
F	Frequency for relays
DF	Frequency Delta (df/dt) for relays
PA:Z	Apparent Impedance for relays
PA:ZH	Apparent Impedance Angle for relays
S	Status Flag for relays
Control Panel Log	Control Panel for Remote Tripe Status
Relay Log	Relay Status (R1 – R4)
Snort Log	Snort Alert Status (R1 – R4)
Marker	Target Column with Label (Triple Class)

Table 2. SAML-Triple with event scenario split for triple class.

Dataset Used			**3-Bus/2-Line Transmission System** [34]
Types of Scenarios	Multiclass Labels	Binary Class	**Triple Class**
Normal Operation	41	Normal	No Events
Single Line-to-Ground Fault	1 to 6		Natural Events
Line Maintenance	13, 14		
False Data Injection Attack	7 to 12	Attack	Attack Events
Remote Tripping Command Injection Attack	15 to 20		
Relay Setting Change Attack	21 to 30, 35 to 40		

3.2. Various Types of Operational Scenarios

- **SLG or Short-circuit fault.** A short circuit in the power line can occur anywhere along the line; the percentage range specifies the fault location.
- **Line maintenance.** This event category is performed when one or more relays are disabled on a specific line so maintenance can be completed for that line.
- **FDI Attack.** Here, the intruder imitates a valid fault by changing values of parameters such as current, voltage, sequence components, etc. The intruders mimic the valid SLG fault by synchronizing with the phasor measurements, followed by sending an illicit trip command to relays at the ends of the transmission line. This attack involves altering the parameters of current, voltage, phase angles, sequence components, and so on to blind the operator without raising the alarm and inducing a blackout. This attack imposes a physical or large-scale load loss and substantial economic loss. Similar to SLG faults, faults can occur at any location in the transmission line with various percentage ranges (10–19%, 20–79%, and 80–90%).
- **Remote tripping command Injection attack.** This attack type arises when an intruder's system on the communications network sends unauthorized relay trip commands to relays at the terminals of a transmission line. The command injection attacks are performed against a single relay (R1 to R4) or two relays (R1 and R2 or R3 and R4).
- **Relay setting change attack.** The intruders alter the relay settings in a distance protection scheme to cause a malfunction in the relay operation. This type of attack fails to recognize valid faults or commands. The faults can occur in any location on the transmission line with R1/R2/R3/R4 disabled and faults.

In this framework [34], a PMU device estimates the magnitude and phase angle of a phasor quantity (such as voltage or current) by synchronizing with a common time source. Each of the four PMU/relays is integrated and measures 29 features, each constituting 116 PMU measurement columns. Following the PMU measurement columns, there are 12 columns for control panel logs, snort alerts, relay logs, and the marker/target column.

4. Methodology

Neighborhood Components Analysis [37] is a supervised non-parametric statistical feature extraction technique based on the K-Nearest Neighbor (KNN) method. KNN makes classifications by grouping an individual data point with the distance between two points using the closest neighbor with a given Euclidean distance (default) metric. The **drawbacks** of **KNN** are that it is a computationally **memory-expensive training** and modeling problem to evaluate the data for a larger dataset, which becomes a **lazy learner**. On the other hand, NCA uses ***Mahalanobis distance*** as a distance measure to optimize (maximize) the selection accuracy and minimize the leave-one-out (LOO) classification error on the training dataset using a stochastic nearest neighbor approach, which **reduces** the **prediction complexity**. It classifies any given test well and speeds up the procedure for faster discrimination. The computation complexity comparison is described in detail under Section 4.3.

4.1. Comparison of Feature Extraction Techniques

- Both NCA and PCA are linearly transformed to lower dimensionality.
- NCA [37] (Supervised Learning) of the statistical approach is a feature extraction technique that employs a method similar to k-Nearest Neighbor in which the neighborhoods of records with the same labels are packed together more densely than those with different labels.
- PCA [27] (Unsupervised Learning) of the statistical approach is a feature extraction technique that projects the matrix into a linear space of lower dimensionality. It transforms a set of correlated variables into a new set of uncorrelated variables called principal components. These principal components are sorted in descending order of variance, capturing the maximum amount of information from the original data in the first few components.
- NCA takes this further by clustering data based on the matrix's dimensionality reduction results with the label.
- **Overall, NCA aims to optimize the selection of local neighborhood relationships for maximum classification accuracy, whereas PCA focuses on capturing global variance without optimization.**

4.2. Neighborhood Components Analysis in the Context of Power Domain

NCA [37] effectively classifies multivariate data into different classes based on the Mahalanobis distance metric computed across the data, based on the distance between a data point and a distribution. It is beneficial for multivariate anomaly detection and classification on highly imbalanced datasets when a correlation between distinct groups or clusters of data is required. It works well when two or more features have high correlations and different scales of values. It matches the power domain context of the ICS Cyber Attack Triple-Class dataset taken with the relationship between associated features of Voltage (V) and Current (I).

NCA accomplishes the same goal as the K-Nearest Neighbor algorithm with a different distance measure of **Mahalanobis distance**, employing stochastic nearest neighbors selection approach by maximizing the selection accuracy and minimizing the training data's leave-one-out (LOO) classification error. In the power domain context of cyberattack detection, the analysis combines multiple dependent variables (features) to predict the classification's single outcome (target column). In SAML-Triple, the dataset contains multivariant data with the features of Voltage Phase Angle and Magnitude, Current Phase Angle and Magnitude, Frequency for relays, Frequency Delta (df/dt) for relays, Appearance

Impedance and its angle for relays and logs feature results in predicting the single outcome (target column) of attack discrimination and classification.

For the **triple-class dataset** taken, let **'X'** represent the matrix of the original feature vector of the dataset. Each row of **'X'** is denoted as x_i corresponds to the feature vector of a single data point. The matrix **'A'** represents the linear transformation applied to the original feature vectors to obtain transformed feature vectors. NCA **aims** to optimize a linear transformation matrix **'A'** for the triple-class dataset. The dataset comprises data records categorized into three distinct classes, each represented by feature vectors. Through iterative optimization with ***conjugate gradient descent***, NCA aims to learn the matrix **'A'** that maximizes the probability of accurate classification in a transformed feature space. This transformation enhances the discrimination between classes, facilitating more effective classification. The gradient-based optimization process involves updating the elements of **'A'** to maximize a predefined cost function, typically measuring the preservation of nearest neighbor relationships in the transformed space. Therefore, maximizing the objective function corresponds to minimizing the leave-one-out (LOO) classification error, leading to better performance of the NCA algorithm on the triple-class dataset. Ultimately, the resulting matrix **'A'** encapsulates the learned transformation, enabling improved classification performance on the triple-class dataset.

Let, p_{ij} be the probability that point x_j is selected as a point x_i's neighbor. The probability that points are correctly classified when x_i is used as a reference is

$$p_i = \sum_{j \in C_i} p_{ij} \text{ where, } C_i = \left\{ x_j \, \text{class}\left(x_j\right) = \text{class}(x_i) \right\} \tag{1}$$

To maximize the p_i of (1) for all x_i means to minimize LOO error. Then, p_{ij} of (2) is defined using the softmax function of the squared Euclidean distance between a given LOO classification point and every other point in the transformed space:

$$p_{ij} = \frac{e^{-d_{ij}}}{\sum_{k \neq i} e^{-d_{ik}}} \quad , p_{ii} = 0 \tag{2}$$

where d_{ij} is defined as a distance measure with between points x_i and x_j provided by

$$\begin{aligned} \boldsymbol{d}_{ij} &= \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) \\ &= (x - y)^T Q (x - y) \\ &= (Ax - Ay)^T (Ax - Ay) \\ &= \|\boldsymbol{Ax_i} - \boldsymbol{Ax_j}\|^2 \end{aligned} \tag{3}$$

Here, x and y are the original feature vectors of matrix **'X'** projected into another vector space with the transformation matrix **'A'**, where **'Q'** is a symmetric, positive semi-definite covariance matrix of the transformed feature space, represented as **'Q'**= $\mathrm{A^T A}$. Here, x_i and x_j , are the transformed feature vectors obtained by multiplying the original feature vectors x and y by the transformation matrix **'A'**, i.e., Ax_i and Ax_j, respectively.

Substituting (3), d_{ij} in (2), p_{ij} is represented below in (4):

$$p_{ij} = \frac{\exp\left(-\left\|Ax_i - Ax_j\right\|\right)^2}{\sum_{k \neq i} \exp\left(-\|Ax_i - Ax_k\|\right)^2}, \text{ where } p_{ii} = 0 \tag{4}$$

Now, the **objective function** of NCA is maximized using LOO classification with stochastic neighbor selection rule, which can be expressed in terms of the cost function $f(A)$ in (5). This cost function $f(A)$ pulls points from the same class closer together.

$$\boldsymbol{f}(\boldsymbol{A}) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_{i=1}^{n} p_i \tag{5}$$

where C_i is the class label of sample i. Since the cost function is differentiable concerning A, it is optimized using gradient descent. Maximizing the objective function $f(A)$ using a **"gradient-based optimizer"** such as **"conjugate gradient descent"** is represented below in (6):

$$\frac{\partial y}{\partial x} = 2A \sum_{i=1}^{n} \left(p_i \sum_{k=1}^{n} p_{ik} x_{ik} x_{ik}^T - \sum_{j \in C_i} p_{ij} x_{ij} x_{ij}^T \right) \quad (6)$$

The terms used in (6) are presented below

- $\frac{\partial y}{\partial x}$ represents the gradient of the objective function $f(A)$ with respect to the elements of the transformation matrix **'A'**.
- n is the total number of samples.
- p_i is the probability associated with sample i.
- p_{ik} is the probability that sample i belongs to class k.
- x_i represents the feature vector of sample i.
- x_{ij} is the j-th element of the feature vector of sample i.
- C_i is the class label of sample i.
- y represents the output of the transformation Ax.

The larger the $f(A)$ during training, the better the test performance. The convergence criteria in NCA are met by observing the changes in the objective function across iterations and stopping the optimization process when the change becomes negligible or the threshold is reached, in addition to setting the maximum number of iterations to prevent the algorithm from running indefinitely.

In the **smart grid context**, NCA is utilized to detect or discriminate cyberattacks in ICS power system datasets; the transformation matrix **'A'** plays a crucial role in mapping the input features to a lower-dimensional space where attack detection or discrimination tasks are performed. In this scenario, the input features from the ICS power system cyberattack dataset, such as voltage, current, phasor values of voltage and current, and frequency, etc., of all 128 features considered are represented in Table 1 as feature description. The transformation matrix **'A'** is utilized to transform these high-dimensional input features into a lower-dimensional space, where the inherent structure of the data relevant to attack detection or discrimination is captured better. The parameters of the transformation matrix **'A'** correspond to the individual elements of the matrix denoted as A_{ij}, which determine the contribution of each input feature to each dimension of the reduced space. During the optimization process in NCA, these parameters are adjusted iteratively to maximize the discriminative power of the reduced space for discriminating cyberattacks from natural and normal instances in the power system data. This optimization typically involves using gradient-based optimization algorithms to update the parameters of **'A'** iteratively, aiming to maximize the objective function that quantifies the effectiveness of the reduced space in detecting or discriminating cyberattacks.

The final values of these parameters after convergence represent the optimized transformation matrix **'A'**, which provides an effective representation of the power system data for attack detection or discrimination purposes.

The **significance of the proposed SAML-Triple approach** is that it utilizes NCA, a supervised non-parametric statistical feature extraction (dimensionality reduction technique), which considers the **local neighborhood structure** relationship between data points belonging to each of three classes (No Events, Natural Events, and Attack Events) and employs Mahalanobis distance measure (3) and the transformation matrix **'A'** to transform into a newer dimension preserving the **global variance** of the data along the principal components. The probability of selecting the closest data points to each class (1) is calculated using the SoftMax function (2) or (4). It leverages class labels to learn a transformation that not only reduces dimensionality but also enhances discriminative information for classification tasks. The objective function of NCA (5) is maximized using

LOO classification with stochastic nearest neighbors selection approach. So, it pulls points from the same class closer together for each of the three classes.

Further, the cost function (5) is optimized using conjugate gradient descent (6) for a better training process for faster convergence. The algorithm aims to learn a **linear transformation** represented by the weight matrix **'A'** that improves the accuracy of the k-Nearest Neighbor classifier (**non-linear**) with Mahalanobis distance measure on the training data. Therefore, NCA effectively maintains both the local and global variance associations of the data, making it particularly well-suited for extracting features from complex datasets that exhibit both linear and non-linear dependencies. The optimal number of components from NCA is hyperparameter tuning with ML classifier parameters, which helps for clear discrimination between Attack Events and Natural Events and No Events.

4.3. Comparison of Computation Complexity

The computational efficiency of both KNN and NCA algorithms involves two phases: training complexity and prediction complexity.

In the case of the **KNN** algorithm, it stores all the training instances (n) in the memory and requires a memory-intensive training phase, and its prediction complexity increases with the number of training instances and the dimensionality of the feature space (d), which is less scalable for large datasets due to **curse of dimensionality**. For each test instance, KNN calculates the distances to all training instances, which requires computing the distance metric (**Euclidean distance** by default) between the test instance and each training instance. The memory complexity of KNN is $O(n * d)$ since all training instances need to be stored in memory. This KNN algorithm comes under the family of **lazy learners**. Beyond a point, KNN is not effective in the discrimination of attacks due to the **Euclidean distance** metric.

NCA involves a more computationally intensive training phase due to a gradient-based optimization algorithm used to learn the transformation matrix **'A'**. The computational complexity per iteration depends on the number of training instances (n) and the dimensionality of the feature space (d). Once the transformation matrix A is learned, making predictions with NCA involves applying this transformation to new instances, which have a computational complexity of $O(d * m)$, where 'm' is the reduced dimensionality. NCA uses **Mahalanobis distance** as a distance measure to optimize (maximize) the selection accuracy in the training phase using the stochastic nearest neighbor approach and minimizing the leave-one-out (LOO) classification error while offering **lower prediction complexity** and better **generalization performance** by learning a transformation to optimize a specified objective function. The memory complexity of NCA depends on the size of the transformation matrix **'A'**, which is typically $O(d * m)$, where 'd' is the original dimensionality and 'm' is the reduced dimensionality.

Overall, the **NCA** is better in **prediction complexity** compared to **KNN**. KNN requires a more memory-intensive training phase, whereas NCA requires more computationally intensive training. Comparatively, NCA achieved better results in terms of **accuracy** in discriminating the cyberattack, and it outperforms KNN. Using the **Mahalanobis distance** in the NCA algorithm further improves accuracy. This has also been demonstrated through our results provided below.

The complexity results for the proposed **SAML approach** using **NCA** achieved 97.51% accuracy compared to KNN with 90.64% accuracy, which shows an **improvement** of **6.87%** accuracy in the classification of attack, which is represented through the line graph in Figure 3.

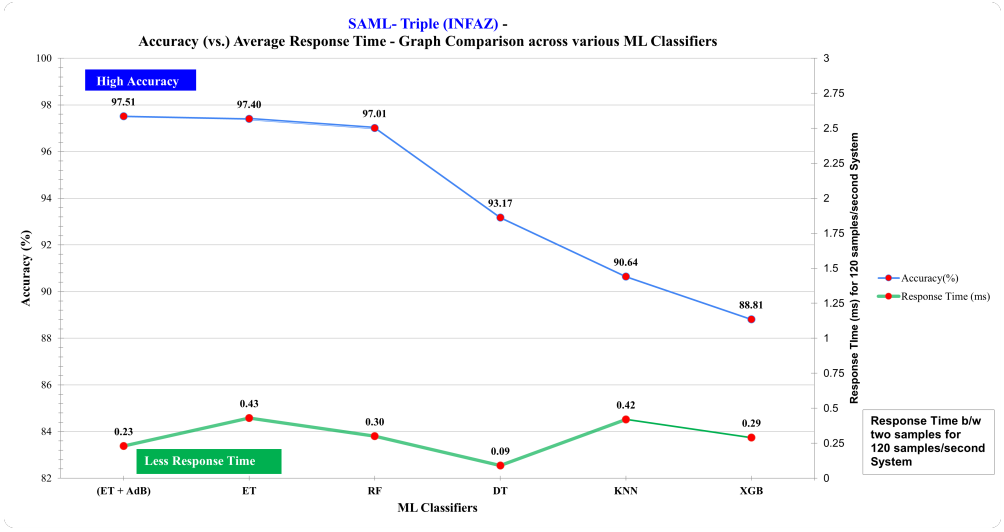


Figure 3. SAML-Triple (INFAZ)—accuracy vs. response time comparison across ML classifiers.

4.4. Proposed Algorithm-SAML

Algorithm 1 depicts the pseudocode for data preprocessing and NCA transformation. The input for Algorithm 1 consists of triple class for SAML-Triple. The output of Algorithm 1 is used to obtain the fitted N-Component List (*N-CompList*) and Iteration List (*ItrList*) for the given datasets. For SAML-Triple, it is carried out with an 80:20 train–test ratio. For SAML-Triple, **Step 1** reads the input dataset into a data frame, which iterates separately for 15 individual datasets. In **Step 2**, the data wrangling is carried out with INFAZ and INFAD. In **Step 3**, the features were split into independent columns (X) and the dependent column as the target column (y). **Step 4** involves standardization (standard scalar) to bring down all the features to a standard scale without distorting the difference in the range of values. In **Step 5**, a label encoder applies the target/marker column as y_label. In **Step 6**, SMOTE is applied for X′ and y_label to balance the dataset into X″, y″. **Step 7** used stratified sampling to perform a train–test split with equal samples from each class. **Steps 8** and **9** involve performing fit and transform on the train data and transform on the test data, respectively, with the ranges specified in Table 3. Finally, **Step 10** stores the train and test records in the pickle format fitted and transformed with the Optimal N-*CompList* and *ItrList*. The pickle format stores the object in the file in byte format and can reload whenever necessary.

Algorithm 2 depicts the Pseudocode for Optimal Hyperparameter tuning to find the *'N' Component* (N) and *Iteration* (I) of NCA with the best parameters for each of the ML classifiers applied. The ML classifiers (C_i) pool is used to evaluate the performance with (**ET + AdB**) ML classifier, ET, DT, RF, KNN, and XGBoost. The input for **Algorithm 2** was obtained from the output of Algorithm 1 with N-*CompList* and *ItrList* of stored NCA train and test in the pickle format. The output includes the optimal hyperparameters for each classifier applied to classify Attack Events from Natural Events and No Events. In **Step 1** of SAML-Triple, the stored pickle NCA data (train and test) are loaded and assigned to NCA_X_train, NCA_y_train, NCA_X_test, and NCA_y_test, respectively, for each of the 15 datasets separately. **Step 2** is used to find the Optimal *'N' Component* (N) and *Iteration* (I) of NCA through GridSearchCV (10-fold cross-validation) to exhaustively search for the best parameters from the given specified parameters provided in Table 3 separately for each of the ML classifiers applied. In **Step 3**, find the best hyperparams from the trained model on each ML classifier (C_i) applied. **Step 4** is used to perform the predictions on the test data with various performance metrics on each of the ML classifiers (C_i) applied. Finally, the test result prediction is used to discriminate the Attack Events from Natural Events and No Events in the triple class.

Algorithm 1 Pseudocode for Data Preprocessing and NCA Transformation

INPUT : ICS/SCADA Cyberattack 15 Datasets of **TRIPLE** Class with 80:20 train–test split
OUPUT : N-Component List ($N - CompList$),
Iteration List ($ItrList$) of storing NCA Train and Test Data

BEGIN
for i : 0 to 14 **do**
1. Dataframe ← Reading dataset
2. INFAZ/INFAD ← Replacing with Zero/Dropping "**INF**inity" Attack Events records
3. X, y ← Splitting input features (X) and "Marker"/Target Label (y)
4. X′ ← Standard Scalar
5. y_label ← Applying Label Encoder
6. X″, y″ ← Applying SMOTE on (X′, y_label)
7. (X_train, y_train) ← Train–Test Split of X″, y″ with Stratisfied Samplling
(X_test, y_test)
for i : product of $N - CompList$ and $ItrList$ with ranges specified
8. NCA_train ← NCA fit and transform on train data (X_train, y_train)
9. NCA_test ← NCA transform on test data (X_test, y_test)
10. Storing NCA_X_train, NCA_y_train, NCA_X_test, NCA_y_test
end
end
END

Table 3. Parameter specifications and optimal hyperparameter tuning of SAML-Triple with NCA.

ML Classifiers	Parameter Specifications	INFAZ (On Average)	INFAD (On Average)
ExtraTrees with AdaBoost Classifier (ET + AdB)	learn. rate = {0.001,0.01,0.1,1}, Base classifier = ExtraTrees, N-Estimator = {50 to 100}, GridSerachCV = 10, Random State = 3	N-Comp = 31 Iterations = 7 learn. rate = 0.001 N-Estimator = 95	N-Comp = 29 Iterations = 6 learn. rate = 0.001 N-Estimator = 95
ExtraTrees (ET)	N-Estimator = {50 to 100}, Criterion = 'entropy', GridSerachCV= 10, Random State = 3	N-Comp = 26 Iterations = 5 N-Estimator = 85	N-Comp = 29 Iterations = 6 N-Estimator = 81
Random Forest (RF)	N-Estimator = {50 to 100}, Criterion = 'entropy', GridSerachCV= 10, Random State = 3	N-Comp = 27 Iterations = 7 N-Estimator = 81	N-Comp = 28 Iterations = 6 N-Estimator = 86
Decision Tree (DT)	Max Depth = {50 to 100}, Criterion= 'entropy', GridSerachCV=10, Random State = 3	N-Comp = 26 Iterations = 6 Max Depth = 95	N-Comp = 28 Iterations = 8 Max Depth = 95
K-Nearest Neighbor (KNN)	K Neighbors = {15 to 70}, GridSearchCV = 10, Random State = 3	N-Comp = 28 Iterations = 8 K Neighbors = 15	N-Comp = 28 Iterations = 9 K Neighbors = 15
Extreme Gradient Boosting (XGB)	learn. rate = {0.001,0.01,0.1}, N-Estimator = {50 to 100}, GridSerachCV = 10, Random State =3	N-Comp = 33 Iterations = 6 learn. rate = 0.1 N-Estimator = 100	N-Comp = 34 Iterations = 7 learn. rate = 0.1 N-Estimator = 100

Algorithm 2 Pseudocode for Optimal Hyperparameter tuning to find the **'N'** *Component* and *Iteration* **'I'** of NCA with the best parameters for each (C_i) classifier applied

INPUT : $N - CompList$, $ItrList$ of stored NCA Train and Test Data in pickle format
OUPUT : Discrimination of Attack Events from Natural Events and No Events in **TRIPLE** Class
BEGIN
Classifiers C = [C_i , $i \epsilon$ 1 to n]
for i : 0 to 14 **do**
1. (NCA_X _train, NCA_y _train, NCA_X _test, NCA_y _test) ← Reading stored NCA data
2. Finding the Optimal 'N' *Component* and *Iteration* 'I' of NCA ← GridSearchCV(k=10 fold) Analysis on each ML (C_i) applied.
3. Finding the best params from each of the trained ML models ← Training on each of the ML classifiers (C_i) for tuning the parameters with the ranges specified.
4. Comparing Performance Metrics ← Predictions on the Test Data with the trained various ML classifiers (C_i).

end
END

A pool of machine learning classifier algorithms [35] used for training and testing the datasets are (**ET + AdB**), ET, RF, DT, KNN, and XGB. For the proposed approach of **SAML-Triple**, the classification algorithm applied is the ExtraTrees (bagging) classifier as a base classifier with the AdaBoost (boosting) technique. ExtraTrees classifier works conceptually like Random Forest but differs in the construction of decision trees. ExtraTrees classifier uses a random selection of features and thresholds at each decision tree with an initial training sample and aggregates the output of multiple decision trees as a "forest". This randomness makes the ExtraTrees classifier less prone to overfitting. Based on the Entropy Index's mathematical criteria, the data partition with the best features is constructed for each decision tree. Further, the AdaBoost technique is applied to transform weak learners into stronger ones by reassigning weights to each incorrectly classified instance.

5. Implementation Details

The proposed approach of **SAML-Triple** is implemented to discriminate triple-class events such as No Events, Natural Events, and Attack Events.

Table 4 represents SAML-Triple (INFAZ)—SMOTE records (before and after) for the 15 datasets. Each of the 15 datasets is highly imbalanced, with the uneven distribution of Attack Events records from 64.73% to 78.13% and normal records (Natural Events and No Events) varying from 21.87% to 35.27% of the distribution. The imbalanced original dataset (without SMOTE) constitutes 4405 rows of No Events records, 18309 rows of Natural Events records, and 55,663 rows of Attack Events records, totaling 78,377 of the triple-class datasets. Meanwhile, with SMOTE (balanced dataset), all three classes are equal to Attack Events records of 55,663, constituting 166,989 records. As per the Pareto principle, an 80:20 ratio of train–test split is performed for both the before and after SMOTE process cases. The original dataset (without SMOTE) constitutes 62,695 records of training samples and 15,682 records of testing samples. SMOTE (balanced dataset) constitutes 133,587 records of training samples and 33,402 records of testing samples.

Table 5 represents preprocessing aspects, train–test split ratio, and SMOTE records (before and after). The preprocessing aspects for SAML-Triple are shown with an example for the first dataset out of fifteen datasets. A similar process is followed for the rest of the datasets.

Table 4. SAML-Triple (INFAZ) before and after SMOTE with the train–test split of the 15 datasets.

Dataset Used	**3-Bus/2-Line Transmission System** [34]											
Feature Engineering	**Without SMOTE (Original Dataset, Imbalanced)**						**With SMOTE (Balanced Dataset)**					
Event Types/ Datasets	**No Events Records**	**Natural Events Records**	**Attack Events Records**	**Total Records (1+2) (100%)**	**Training Sample (1) (80%)**	**Testing Sample (2) (20%)**	**No Events Records**	**Natural Events Records**	**Attack Events Records**	**Total Records (1+2) (100%)**	**Training Sample (1) (80%)**	**Testing Sample (2) (20%)**
Dataset 1	173	927	3866	4966	3972	994	3866	3866	3866	11,598	9278	2320
Dataset 2	322	1222	3525	5069	4055	1014	3525	3525	3525	10,575	8460	2115
Dataset 3	354	1250	3811	5415	4332	1083	3811	3811	3811	11,433	9146	2287
Dataset 4	403	1397	3402	5202	4161	1041	3402	3402	3402	10,206	8164	2042
Dataset 5	270	1211	3680	5161	4128	1033	3680	3680	3680	11,040	8832	2208
Dataset 6	190	1287	3490	4967	3973	994	3490	3490	3490	10,470	8376	2094
Dataset 7	208	1118	3910	5236	4188	1048	3910	3910	3910	11,730	9384	2346
Dataset 8	356	1188	3771	5315	4252	1063	3771	3771	3771	11,313	9050	2263
Dataset 9	478	1292	3570	5340	4272	1068	3570	3570	3570	10,710	8568	2142
Dataset 10	326	1322	3921	5569	4455	1114	3921	3921	3921	11,763	9410	2353
Dataset 11	145	1137	3969	5251	4200	1051	3969	3969	3969	11,907	9525	2382
Dataset 12	384	1387	3453	5224	4179	1045	3453	3453	3453	10,359	8287	2072
Dataset 13	203	950	4118	5271	4216	1055	4118	4118	4118	12,354	9883	2471
Dataset 14	79	1274	3762	5115	4092	1023	3762	3762	3762	11,286	9028	2258
Dataset 15	514	1347	3415	5276	4220	1056	3415	3415	3415	10,245	8196	2049
Total	4405	18,309	55,663	**78,377**	**62,695**	**15,682**	55,663	55,663	55,663	**166,989**	**133,587**	**33,402**

Table 5. SAML-Triple with preprocessing aspects and train–test split ratio before and after SMOTE records for the first dataset.

Dataset Used		**3-Bus/2-Line Transmission System** [34]							
Preprocessing Aspects		**INFAZ**				**INFAD**			
		Applying SMOTE on each of the 15 Datasets Separately (e.g., Dataset-1)				**Applying SMOTE on Each of the 15 Datasets Separately (e.g., Dataset-1)**			
Train–Test Split		80:20 ratio							
Event Types		No Events records	Natural Events records	Attack Events records	**Total Records**	No Events records	Natural Events records	Attack Events records	**Total Records**
Before SMOTE		173	927	3866	4966	173	835	3610	4618
After SMOTE	**Training Samples**	3093	3093	3092	9278	2888	2888	2888	8664
	Testing Samples	774	773	773	2320	722	722	722	2166
No. of records		3866	3866	3866	**11,598**	3610	3610	3610	**10,830**

Table 6 represents the terminology used in the specifications of the parameters.

Table 6. The terminology used in the specifications of the parameters.

Name of the Parameter	**Meaning**
N-Estimator	Number of trees in the forest
Criterion	The quality measure of the split

Table 6. *Cont.*

Name of the Parameter	Meaning
Cross-validation	Stratified K-Fold Cross-validation
Random State	Controls the randomness of the estimator
Max Depth	The maximum depth of the tree.
classifier	Base classifier to be specified
learning_rate	Weight is applied to each classifier at each boosting iteration.
K	Number of nearest neighbors

Table 7 represents the parameter specifications of the feature extraction technique (NCA) with the specified range to find the optimal 'N' *Component* and maximum *Iteration* 'I'. The parameter specification range for SAML-Triple lies between 20 and 35 components in the NCA component list, and iterations range from 2 to 10 with a two-step increment. The range is set based on trials and broad studies in related work in Section 2.

Table 7. Parameter specifications of feature extraction technique (NCA).

Dataset Used	3-Bus/2-Line Transmission System [34] Individual 15 Datasets
Feature Extraction Technique of NCA	NCA components list = [20,25,30,35] Max iterations = [2,4,6,8,10]

Table 3 represents the parameter specifications and optimal hyperparameter tuning of the SAML-Triple with NCA. For the SAML-Triple, optimal hyperparameter tuning results were obtained from Table 8 on an average of 15 datasets, and similar approaches were performed for other ML classifiers to obtain the same.

Table 8. SAML-Triple—performance metrics comparison of NCA with AdaBoost (ET*) classifiers in two preprocessing aspects.

DS	Preprocessing as INFAZ								Preprocessing as INFAD							
	N-Comp	Iter.	Learning Rate	N-Esti.	Prec.	Recall	F1-Score	Acc.	N-Comp	Iter.	Learning Rate	N-Esti.	Prec.	Recall	F1-Score	Acc.
1	35	6	0.001	95	98.84	98.84	98.84	**98.84**	30	4	0.001	95	98.48	98.48	98.48	**98.48**
2	20	8	0.001	95	97.26	97.26	97.26	**97.26**	30	10	0.001	95	97.97	97.97	97.97	**97.97**
3	35	8	0.001	95	97.86	97.86	97.86	**97.86**	35	2	0.001	95	98.72	98.72	98.72	**98.72**
4	35	6	0.001	95	97.16	97.16	97.16	**97.16**	20	10	0.001	95	98.40	98.40	98.40	**98.40**
5	35	8	0.001	95	97.61	97.60	97.60	**97.60**	35	2	0.001	95	98.13	98.13	98.13	**98.13**
6	30	8	0.001	95	96.94	96.94	96.94	**96.94**	20	6	0.001	95	97.39	97.37	97.37	**97.37**
7	35	8	0.001	95	97.66	97.66	97.66	**97.66**	25	4	0.001	95	97.93	97.93	97.93	**97.93**
8	25	4	0.001	95	97.17	97.17	97.17	**97.17**	30	6	0.001	95	99.10	99.10	99.10	**99.10**
9	25	8	0.001	95	96.60	96.59	96.60	**96.59**	30	6	0.001	95	97.88	97.88	97.88	**97.88**
10	35	4	0.001	95	97.64	97.62	97.62	**97.62**	30	2	0.001	95	98.33	98.33	98.33	**98.33**
11	30	4	0.001	95	97.86	97.86	97.86	**97.86**	35	6	0.001	95	98.10	98.10	98.10	**98.10**
12	25	8	0.001	95	96.52	96.48	96.49	**96.48**	35	10	0.001	95	98.20	98.20	98.20	**98.20**
13	35	8	0.001	95	98.63	98.62	98.63	**98.62**	35	4	0.001	95	99.22	99.21	99.21	**99.21**
14	35	10	0.001	95	97.79	97.79	97.79	**97.79**	30	10	0.001	95	98.58	98.57	98.57	**98.57**
15	35	4	0.001	95	97.23	97.22	97.22	**97.22**	20	10	0.001	95	97.38	97.35	97.36	**97.35**
Avg.	**31**	**7**	**0.001**	**95**	97.52	97.51	97.51	**97.51**	**29**	**6**	**0.001**	**95**	98.25	98.25	98.25	**98.25**

Table 9 represents the possible test case scenarios. SAML-Triple has seven test cases with an Attack Events label as (0), Natural Events label as (1), and No Events label as (2).

Table 9. SAML-Triple—possible test case scenarios.

Test Cases ine **Labels**	**Case 1** **0, 1, and 2**	**Case 2** **0**	**Case 3** **1 and 2**	**Case 4** **1**	**Case 5** **0 and 2**	**Case 6** **2**	**Case 7** **0 and 1**
Types of Scenarios	Attack Events vs. Natural Events vs. No Events	Attack Events	Natural Events vs. No Events	Natural Events	Attack Events vs. No Events	No Events	Attack Events vs. Natural Events

Tools for Implementation and Evaluation Metrics

The tool used for implementation is the Online Google Colab Data Analytics platform (free subscription). It uses Python 3 Google Compute Engine utilizing 13 GB RAM, a 2-core Xeon CPU @ 2.20 GHz, a processor, and 108 GB of hard disks.

SAML-Triple utilizes the NCA algorithm that requires this configuration to find the optimal 'N' *Component* and maximum *Iteration* 'I' for the specified range, as provided in Table 3. The confusion matrix represented in Figure 4 is used to evaluate the performance of a machine learning model on the testing data for triple class. It summarizes the model's predictions and the actual values for the classification problem. For the performance evaluation of IDS, SAML-Triple adopts standard metrics, such as accuracy, FNR, FPR, response time, precision, recall, and F1-score. Figure 4 represents the confusion matrix of SAML-Triple between actual versus predicted classes. True positive cases for three scenarios (marked in green) $TP_{(AE)}$ is correctly classified as Attack Events, $TP_{(NaE)}$ is correctly classified as Natural Events, and $TP_{(NoE)}$ is correctly classified as No Events, whereas true negatives occur for three cases regarding the respective left and right diagonal elements (marked in red). The false negatives for the three cases were respective horizontal rows, and false positive cases for the three were vertical columns. In the attack scenario case, $F_{(AE\text{-}NaE)}$ and $F_{(AE\text{-}NoE)}$ Attack Events were misclassified as Natural Events and No Events, respectively. In the case of the Natural Events scenario, $F_{(NaE\text{-}AE)}$ and $F_{(NaE\text{-}NoE)}$ were misclassified as Attack Events and No Events. In the case of the No Events scenario, $F_{(NoE\text{-}AE)}$ and $F_{(NoE\text{-}NaE)}$ were misclassified as Attack Events and Natural Events.

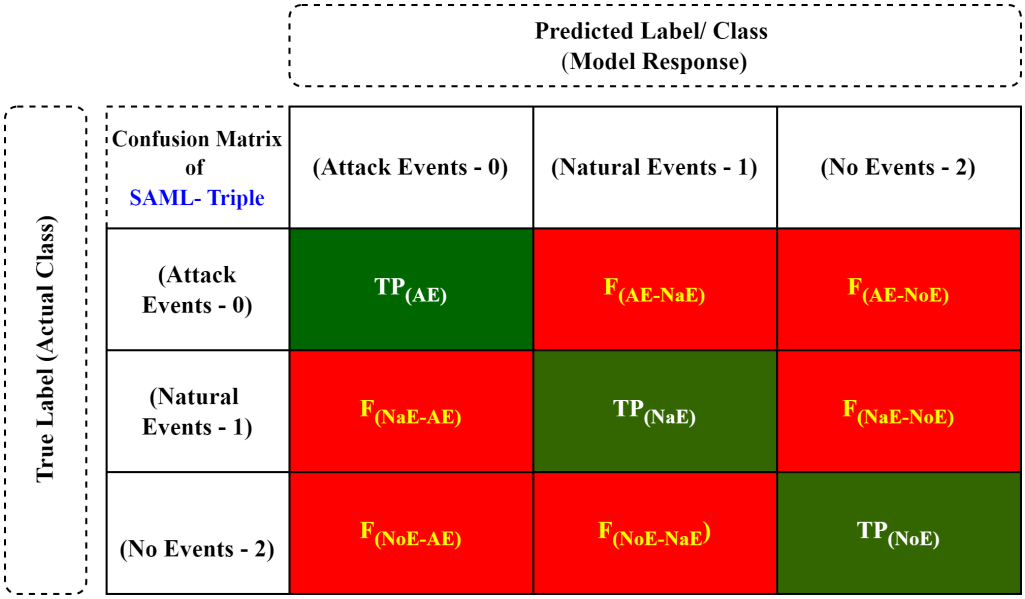


Figure 4. General representation of confusion matrix for SAML-Triple [34].

6. Results Analysis and Discussion

SAML-Triple provides a robust solution for discriminating Attack Events from Natural Events and No Events, represented by tables, graphs, and a confusion matrix.

6.1. SAML-Triple: Triple-Class Datasets (No Events/Natural Events/Attack Events)—Test Results

Table 10 represents the potential impact of the SMOTE operation on datasets. The SMOTE operation achieved higher accuracy for all 15 triple-class datasets with an equal number of records considered from each class through stratified sampling, specified in Table 4 in Section 5, Implementation Details.

Table 10. SAML-Triple (INFAZ)—comparison between without SMOTE and with SMOTE.

Feature Extraction Techniques with ML Classifier	SAML-NCA with (ET + AdB) Classifier (Without SMOTE)	SAML-NCA with (ET + AdB) Classifier (With SMOTE)
Datasets	**Acc. (%)**	**Acc. (%)**
1	97.08	98.84
2	94.08	97.26
3	95.75	97.86
4	96.83	97.16
5	95.45	97.60
6	95.37	96.94
7	95.61	97.66
8	94.92	97.17
9	94.94	96.59
10	96.32	97.62
11	96.00	97.86
12	95.41	96.48
13	97.35	98.62
14	95.60	97.79
15	95.27	97.22
Avg.	**95.73**	**97.51**

From Figure 5, it is indicated that, without SMOTE, the dataset is imbalanced with three label records, showing decreased accuracy for all 15 datasets marked with a blue line, with an average accuracy of 95.73% compared to the SMOTE operation, with an average accuracy of 97.51%, marked with a red line.

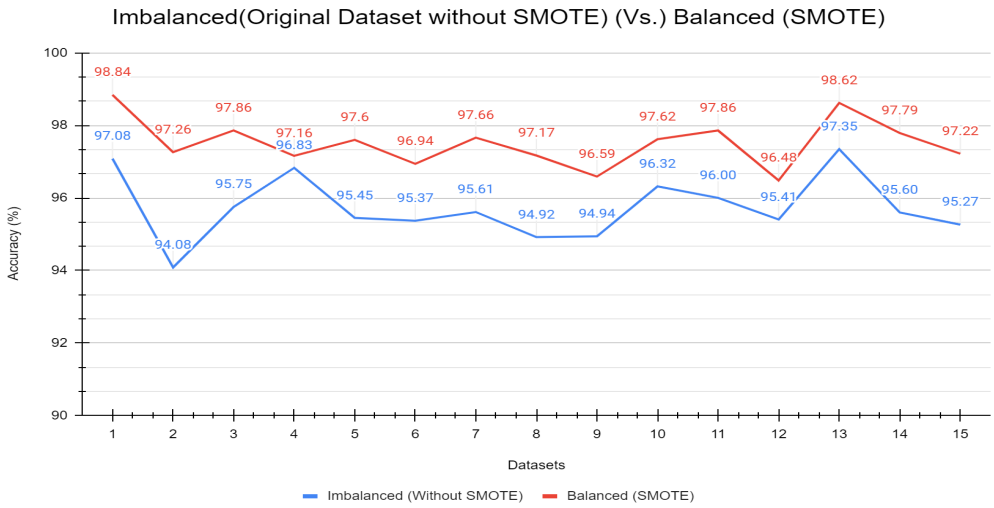


Figure 5. SAML-Triple (INFAZ) graph comparison of imbalanced (original dataset without SMOTE) vs. balanced (with SMOTE).

Table 11 represents the SAML-Triple accuracy metric comparison of feature extraction techniques—NCA vs. PCA with the (ET + AdB) classifier with two preprocessing aspects,

INFAZ and INFAD. The NCA with the (ET + AdB) classifier provides an average higher accuracy of 97.51% and 98.25% for INFAZ and INFAD, respectively, than the PCA with the (ET + AdB) classifier of 97.25% and 97.69%. The accuracy for the SAML-NCA with (ET + AdB) with INFAZ and INFAD was obtained from Table 8, and a similar approach was carried out for the PCA technique.

Table 11. SAML-Triple—comparison of NCA vs. PCA with two preprocessing aspects.

Feature Extraction Techniques with ML Classifier	**PCA [27] with (ET + AdB) Classifier**		**SAML-NCA with (ET + AdB) Classifier**	
Preprocessing Aspects	**INFAZ**	**INFAD**	**INFAZ**	**INFAD**
Datasets	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**
1	98.23	98.34	98.84	98.48
2	97.16	97.21	97.26	97.97
3	97.60	98.06	97.86	98.72
4	96.87	96.64	97.16	98.40
5	96.42	98.08	97.60	98.13
6	97.13	96.34	96.94	97.37
7	97.95	97.93	97.66	97.93
8	97.44	98.43	97.17	99.10
9	96.36	96.66	96.59	97.88
10	97.32	97.63	97.62	98.33
11	97.19	98.10	97.86	98.10
12	96.62	97.67	96.48	98.20
13	98.50	98.78	98.62	99.21
14	97.17	98.14	97.79	98.57
15	96.73	97.30	97.22	97.35
Avg.	97.25	97.69	**97.51**	**98.25**

Figure 6 shows the SAML-Triple accuracy (bar) graph of the NCA vs. PCA with the (ET + AdB) ML classifier with two preprocessing aspects of INFAZ and INFAD. Figure 6 implies that NCA_INFAZ (blue bar) is comparatively higher in accuracy than PCA_INFAZ (yellow bar) across 15 datasets, and the same is true for NCA_INFAD (red bar) vs. PCA_INFAD (green bar). From another perspective, NCA_INFAD (red bar) performs comparatively better than NCA_INFAZ (blue bar), which is not an ideal case that may lead to a missing attack rate; hence, NCA_INFAZ (blue bar) provides good accuracy of more than 96% across 15 datasets.

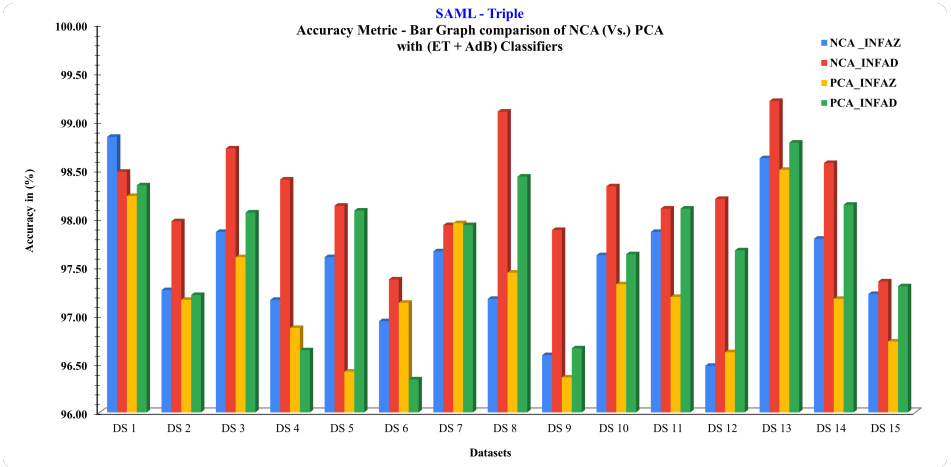
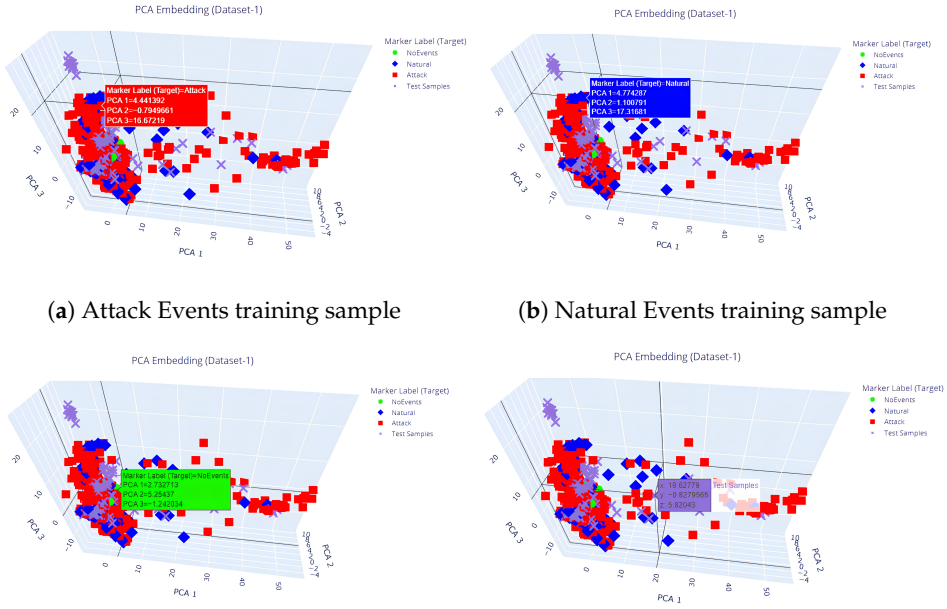


Figure 6. SAML-Triple—accuracy (bar) graph comparison of NCA vs. PCA with ExtraTrees with AdaBoost (ET + AdB) classifier.

Figure 7 depicts the SAML-Triple (INFAZ) approach using the PCA representation of three events with the (ET + AdB) ML classifier (e.g., Dataset-1). The three axes, x, y, and z, represent PCA-1, PCA-2, and PCA-3, respectively. These three axes' values represent the top three components with high variance from the transformed 128 features, which retain the maximum relative information. Figure 7a represents the Attack Events training samples, Figure 7b represents the Natural Events training samples, and Figure 7c represents the No Events training samples. Figure 7d represents the test samples that are tedious to discriminate between the Attack Events and Natural Events that might belong to any of the three events.



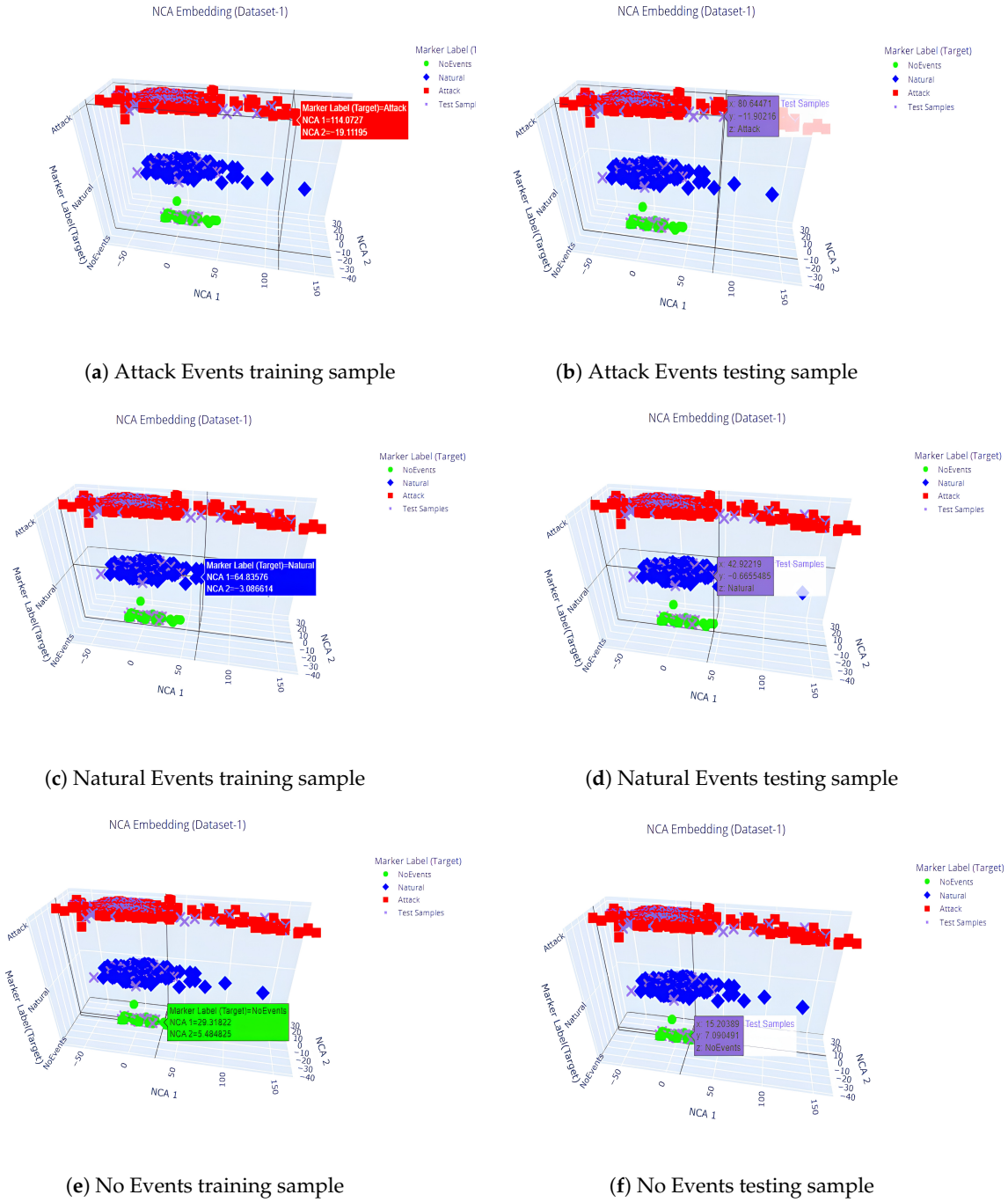
(**a**) Attack Events training sample (**b**) Natural Events training sample

(**c**) No Events training sample (**d**) Testing samples

Figure 7. SAML-Triple (INFAZ) using PCA Representation of (**a**) Attack Events training sample; (**b**) Natural Events training sample; (**c**) No Events training sample; (**d**) testing samples.

Figure 8 depicts the SAML-Triple approach using NCA representation with discrimination of three events using the (ET + AdB) ML classifier (e.g., Dataset-1). The three axes, x, y, and z, represent NCA-1, NCA-2, and marker (target) columns. The first two axes' values represent the top two components with high variance from the transformed 128 features, which retain maximum relative information. In contrast, the third axis is a marker (target)-labeled column. It clearly shows the discrimination of Attack Events, Natural Events, and No Events since the third axis is a marker (target)-labeled column, which aids in discriminating the processed component level values per the mathematical objective defined in Section 4. Figure 8a,b show the training and testing samples of Attack Events. Figure 8c,d show the training and testing samples of Natural Events. Figure 8e,f show No Events' training and testing samples.

Table 8 represents a comparison of the performance metrics of SAML-Triple using NCA with the (ET + AdB) ML classifier in two preprocessing aspects. The testing samples of 20% were taken for evaluation across 15 datasets with INFAZ and INFAD. The accuracy, precision, recall, and F1-score performance metrics were compared. On average, 97.51% and 98.25% were achieved by considering INFAZ and INFAD, respectively. Each of the 15 datasets is executed with optimal hyperparameter tuning to find the optimal 'N' *Component* and maximum *Iteration* 'I' for NCA as a feature extraction technique and optimal 'N' estimator for the (ET + AdB) ML classifier.



(**a**) Attack Events training sample

(**b**) Attack Events testing sample

(**c**) Natural Events training sample

(**d**) Natural Events testing sample

(**e**) No Events training sample

(**f**) No Events testing sample

Figure 8. SAML-Triple (INFAZ) using NCA Representation: (i) discrimination of Attack Events: (**a**) training sample ; (**b**) testing sample; (ii) discrimination of Natural Events: (**c**) training sample; (**d**) testing sample; (iii) discrimination of No Events: (**e**) training sample; (**f**) testing sample.

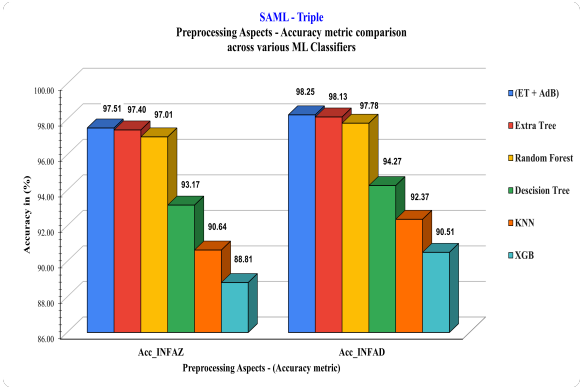
Table 12 compares the SAML-Triple classification accuracy comparison of various test cases with two preprocessing aspects of INFAZ and INFAD. The test results across various test cases achieved more than 96% average accuracy for both the preprocessing aspects.

Table 13 represents the SAML-Triple performance metrics comparison of NCA with the (ET + AdB) ML classifier obtained from Table 8. The results were obtained using a similar approach to the rest of the ML classifiers. The performance metrics of precision, recall, F1-score, accuracy, FPR, FNR, and testing time results were compared with distinct preprocessing aspects of INFAZ and INFAD.

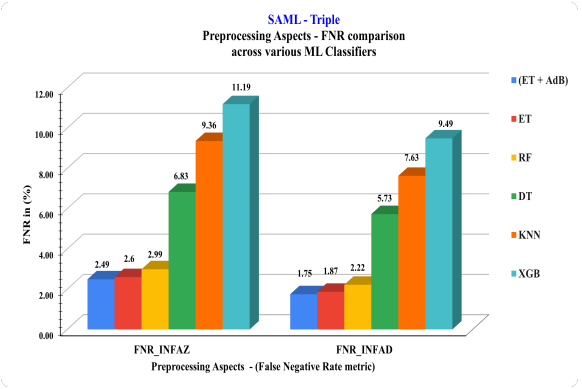
Table 12. SAML-Triple—classification accuracy of various test cases with two preprocessing aspects.

Test Case Scenarios of 15 Datasets	Case 1		Case 2		Case 3		Case 4		Case 5		Case 6		Case 7	
	Attack Events vs. Natural Events vs. No Events		Attack Events		Natural Events vs. No Events		Natural Events		Attack Events vs. No Events		No Events		Attack Events vs. Natural Events	
Average of 15 Datasets Accuracy (%)	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD
	97.51	98.25	96.66	97.82	97.94	98.46	96.40	97.21	98.07	98.77	99.48	99.71	96.53	97.52

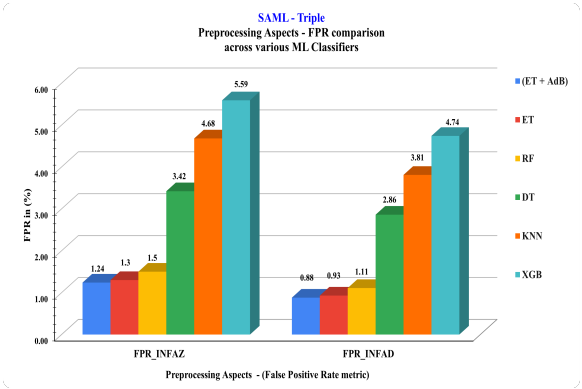
Figure 9a represents the accuracy (bar) graph comparison of SAML-Triple with two preprocessing aspects, INFAZ and INFAD. Figure 9a implies that the (ET + AdB) ML classifier (blue bar) achieved a higher accuracy of 97.51% and 98.52% for INFAZ and INFAD, respectively, than the rest of the ML classifiers. Figure 9b is an FNR graph (missing rate) comparison of SAML-Triple with two preprocessing aspects. Figure 9b implies that the (ET + AdB) ML classifier (blue bar) achieved the lowest FNR values of 2.49% and 1.75% for INFAZ and INFAD, respectively, compared to the rest of the ML classifiers. Figure 9c is an FPR graph (false alarm) comparison of SAML-Triple with two distinct preprocessing aspects. Figure 9c implies that the (ET + AdB) ML classifier (blue bar) achieved the lowest FPR values of 1.24% and 0.88% for INFAZ and INFAD, respectively, compared to the rest of the ML classifiers.



(**a**) Accuracy Graph



(**b**) False Negative Rate



(**c**) False Positive Rate

Figure 9. SAML-Triple—comparison of various ML classifiers with two preprocessing aspects: (**a**) accuracy graph; (**b**) false negative rate; (**c**) false positive rate.

Table 13. SAML-Triple—performance metrics comparison of various ML classifiers with two preprocessing aspects.

Proposed Work vs. Other ML Alg.	Precision (%)		Recall (%)		F1-score (%)		Accuracy (%)		FPR (%)		FNR (%)	
	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD	INFAZ	INFAD
(ET + AdB)	97.52	98.25	97.51	98.25	97.51	98.25	**97.51 ***	**98.25 ***	**1.24**	**0.88**	2.49	1.75
ET	97.41	98.13	97.40	98.13	97.40	98.13	97.40	98.13	1.30	0.93	2.60	1.87
RF	97.02	97.79	97.01	97.78	97.01	97.78	97.01	97.78	1.50	1.11	2.99	2.22
DT	93.17	94.28	93.17	94.28	93.16	94.27	93.17	94.27	3.42	2.86	6.83	5.73
KNN	90.77	92.45	90.65	92.38	90.57	92.32	90.64	92.37	4.68	3.81	9.36	7.63
XGB	88.80	90.50	88.81	90.51	88.75	90.44	88.81	90.51	5.59	4.74	11.19	9.49

*—represents the highest accuracy achieved through proposed approach.

Table 14 represents the SAML-Triple (INFAZ) average response time comparison across various ML classifiers for 120 samples/second system [36]. The average response time was obtained with batch processing of 120 samples/second test records for 10 rounds of batch processing.

Table 14. SAML-Triple (INFAZ)—average response time of various ML classifiers for 120 samples/second system.

10 Rounds of Batch Processing	Response Time (ms) for 120 Samples/s System [36]					
	(ET + AdB)	ET	RF	DT	KNN	XGB
Round-1	31.68	58.72	34.66	10.70	109.62	42.02
Round-2	24.52	46.10	36.70	11.87	41.28	45.61
Round-3	26.20	47.89	35.12	11.52	43.11	47.72
Round-4	26.00	61.02	34.54	10.67	35.06	39.12
Round-5	23.62	59.68	36.43	12.02	38.49	42.25
Round-6	38.23	46.46	44.46	9.77	29.35	40.37
Round-7	39.23	46.57	34.97	9.54	36.67	37.75
Round-8	23.32	47.44	34.44	9.04	35.63	17.68
Round-9	24.07	53.46	33.99	8.74	39.94	18.67
Round-10	23.72	54.09	34.25	11.4	100.04	17.99
Average Response Time (ms)	**28.06**	52.14	35.96	10.53	50.92	34.92

The data generated from the power system framework [34] were collected in the PDC (Phasor Data Concentrator). For a synchrophasor system [36] with 120 samples/second, there are 8.3 ms between samples; this time could be employed to process the samples with the SAML-Triple (INFAZ) approach, and it can detect anomalies with less than 8.3 ms between samples. Table 15 represents the SAML-Triple (INFAZ) accuracy with average response time and per sample response time comparison across various ML classifiers.

The proposed **SAML-Triple (INFAZ)** IDS can process within 0.23 ms between samples and less than **8.3 ms** between samples for a 120 samples/second system [36]. The Attack Events records of "**INF**inity" are processed in the SAML-Triple (INFAZ) approach, whereas other existing approaches are lagging in this aspect, as discussed in Section 2. If the Attack Events records are unprocessed (INFAD), it might have fatal consequences for the stability of the power system and cause the system to collapse. The SAML-Triple (INFAZ) approach, with a response time of 0.23 ms per sample, can alert the prevention system to take further

action to regain the power system's stability. This crucial response time is more critical in the mission-critical infrastructure of the smart grid to undertake faster decision operations, which is achieved through our proposed work of SAML-Triple (INFAZ).

Table 15. SAML-Triple (INFAZ)—accuracy vs. average response time—graph comparison across various ML classifiers.

Comparison with Various ML Classifiers	Accuracy (%)	Average Response Time (ms) for 120 Samples/s System [36]	Response Time (ms) between two Samples of 8.3 ms [36]
(ET + AdB)	**97.51**	**28.06**	**0.23**
ET	97.40	52.14	0.43
RF	97.01	35.96	0.30
DT	93.17	10.53	0.09
KNN	90.64	50.92	0.42
XGB	88.81	34.92	0.29

Figure 3 has an x-axis with various ML classifiers and two y-axes, with the left y-axis representing the accuracy metric and the right y-axis representing the response time metric. Figure 3 implies that the SAML-Triple (INFAZ) NCA with the (ET + AdB) ML classifier performs better, with an accuracy of 97.51% and a response time of 0.23 ms, detecting an attack that is less than 8.3 ms between samples (120 samples/second) [36]. Even though the DT classifier has a lower response time of 0.09 ms compared to the (ET + AdB) ML classifier of 0.23 ms, the accuracy of the DT classifier remains low at 93.17%.

Figure 10 represents the confusion matrix of SAML-Triple (INFAZ) for the first dataset out of 15 datasets after the SMOTE operation represented in Table 5. The total number of samples in the first dataset before SMOTE is 4966, with 173 (No Events), 927 (Natural Events), and 3866 (Attack Events). After applying SMOTE to the first dataset, the total number of sample records was 11,598, with an equal number of 3866 records for each of the three events (No Events, Natural Events, and Attack Events). Out of the 11,598 records, 9278 were taken for training, and 2320 were taken for testing, with a ratio of 80:20 as per the Pareto principle. Moreover, 80% of the training samples' 9278 records contain almost an equal number of 3093 samples from the three events.

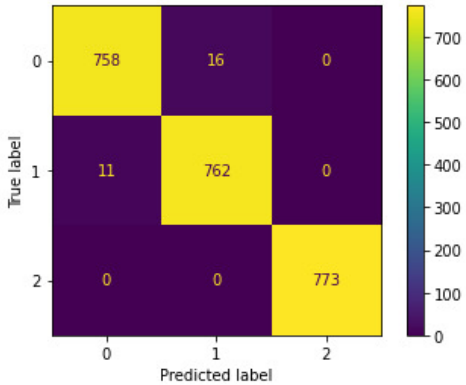


Figure 10. Confusion matrix of SAML-Triple (INFAZ) after SMOTE on testing samples (e.g., Dataset-1).

For the remaining 20% of the testing samples, 2320 records containing almost equal numbers of 774 samples from the three events were considered. With the testing samples of 2320 records, the confusion matrix in Figure 10 depicts the True Label vs. Predicted Label reference to the confusion matrix in Figure 4. The labels in the confusion matrix in Figure 10 represent Attack Events (0), Natural Events (1), and No Events (2). From

the confusion matrix of SAML-Triple (INFAZ) in Figure 10, in the attack scenario case, 758 samples (marked in yellow) were correctly classified as Attack Events (true positive), whereas 16 samples (marked in violet) were misclassified as Natural Events (false negative). In the case of the Natural Events scenario, 762 samples (marked in yellow) were correctly classified as Natural Events (true positive), whereas 11 samples (marked in violet) were misclassified as Attack Events (false negative). In the case of the No Events scenario, 773 samples (marked in yellow) were correctly classified as No Events, and there were no misclassifications. Furthermore, the false positives are the opposite for the three scenarios concerning false negatives.

Table 16 compares the SAML-Triple accuracy metrics of NCA with the (ET + AdB) ML classifier vs. other existing approaches. SAML-Triple achieved a higher accuracy of 97.51% and 98.25% by considering INFAZ and INFAD, respectively. The INFAZ aspect can address the missing rate, whereas the INFAD aspect will not deal with the missing rate. Both aspects of comparison were conducted with the existing approaches with the number of features selected or extracted for classification. The SAML-Triple (INFAZ) approach outperforms the other existing approaches with an accuracy of 97.51%. The proposed approach with 31 components preserves the data's local and global variance associations, making it well-suited for extracting features from highly complex correlated datasets that exhibit linear and non-linear dependencies.

Table 16. Accuracy metric comparison of SAML-Triple with other existing approaches.

Reference Paper	Feature Selection/ Extraction	Number of Features Selection or Extraction	Machine Learning Classifiers	Accuracy (%)	
				INFAZ	INFAD
SAML-Triple (Proposed Work)	**NCA**	**31**	**(ET + AdB)**	**97.51 ***	**98.25 ***
Upadhyay, Darshana, et al. [22,23]	GBFS	15	Tree Based	-	96.50
	RFE-XG	30	MV-EM	-	97.95
Hu, Chengming, et al. [24,25]	Stacked Denoising Autoencoders (SDAE)	60	XGBoost	90.48	
	Multiple Autoencoders (AE)	30	Random Forest	91.78	
Gumaei, Abdu, et al. [26]	Correlation-Based Feature Selection	8 to 11	KNN	91.87	
Ankitdeshpandey, Karthi, R. [27]	PCA	31	Random Forest	91.14	
Hink, Raymond, C. Borges, et al. [28]	Information Gain	40	Adaboost + JRipper	95.00	
Agrawal, Anand, et al. [29]	ExtraTrees	10	LightGBM	-	95.30
Sunku Mohan, Vamshi, and Sriram Sankaran [30]	Manually selected the Features based on Power Domain Knowledge	36	Rule-Based ML + AdaBoost	97.25	
Bitirgen K, Filik ÜB [31]	PSO	Not Specified	CNN-LSTM	96.92	

*—represents the highest accuracy achieved through proposed approach.

Specifically, we addressed the "INFinity" Attack Events records in the feature column of "PA:Z" (Apparent Impedance for Four Relays) by replacing them with "Zero" (INFAZ), which avoids the missing rate, which can maintain the power system's stability and reliability. Meanwhile, the existing approaches [22,23,26,28,29,31] lack the selection of suboptimal features with feature importance scores, leading to **potential feature selection bias**. They lack sufficient performance in discriminating Attack Events from Natural Events and No Events. The rest of the existing approaches [24,25], which use the feature extraction techniques of deep learning methods, lack the optimal combination of extracting the features due to several hyperparameter factors. Moreover, the feature extraction using PCA [27] fails to capture the local structure or relationships within the data, which might result in misclassification between the three classes. Also, the author in [30] undertook manual feature selection, which may not be suitable for the generalizability and scalability of the model for different architectures and may require complex logical calculations. Meanwhile, our SAML-Triple (INFAZ) can be scalable and generalizable to the IEEE 'N' bus system for different architectures.

The limitation of the ICS Cyber Attack Power System Triple-Class Dataset [34] is that the data generated for the No Events records are less than the other two events. Furthermore, the Natural Events records are less than the Attack Events records. Due to the imbalance of the dataset, SMOTE is required to balance it, as depicted in Table 4. Its potential impact is shown in Table 10, with the accuracy metric comparing those without SMOTE vs. with SMOTE.

6.2. SAML-Triple: Generalization and Scalability for IEEE 'N' Bus System

For the generalization and scalability of the proposed approach of SAML-Triple, we have considered the IEEE 14- and 57-bus systems' datasets from J. Sakhnini et al. [13], generated with the MATPOWER library. The dataset is posted publicly as an open source at the GitHub link [38]. The author simulated an FDI attack on the IEEE 14- and 57-bus systems with 10,000 training records and 1000 records for testing on each bus system. The IEEE 14-bus system has 34 feature columns, whereas the IEEE 57-bus system has 137 feature columns.

Table 17 represents the parameter specifications of the feature extraction technique (NCA) with the specified range to find the optimal 'N' component and maximum iteration 'I' for the IEEE 14- and 57-bus systems [38]. The parameter specification range for the IEEE 14-bus system lies between 2 and 10 NCA components, the iteration range from 5 to 20, and a learning rate of 0.001. In contrast, the IEEE 57-bus system lies from 60 to 90 NCA components, with an iteration range of 5 to 20 and a learning rate of 0.001. The choice of range for each bus system is set below the number of actual features.

Table 17. Parameter specifications of feature extraction technique (NCA) for IEEE 'N' bus system.

Dataset Used	No. of Features	NCA Components and Iteration Range
IEEE 14 Bus System [38]	34	NCA components list = [2, 5, 10] Max iterations: [5, 10, 15, 20] learning_rate = {0.001}
IEEE 57 Bus System [38]	137	NCA components list = [60, 70, 80, 90] Max iterations = [5, 10, 15, 20] learning_rate = {0.001}

Table 18 represents the generalization and scalability of the proposed SAML-Triple approach to the IEEE 14- and 15-bus systems with accuracy metric comparison. The IEEE 14- and 57-bus datasets [38] provided by the author [13] used the Binary Cuckoo Search (BCS) optimization algorithm as a Heuristic Feature Selection approach to select the optimal feature subset. BCS is susceptible of converging to local optima, especially in complex and irregular fitness landscapes. It may struggle to select the features due to premature convergence and suboptimal solutions. Identifying suitable parameter values for this dataset [38] may require extensive tuning. This may limit its ability to find globally optimal or near-optimal solutions. Our proposed SAML-Triple approach utilizing NCA preservers the data's local and global variance associations, making it well-suited for extracting features from highly complex correlated datasets that exhibit linear and non-linear dependencies. The detailed significance of the proposed approach is explained in Methodology Section 4.

Table 18. Generalization and scalability of proposed SAML-Triple approach to IEEE 'N' bus systems with accuracy metric comparison.

IEEE 'N' Bus Systems	**IEEE 14-Bus System** [38]		**IEEE 57-Bus System** [38]	
Comparison of Proposed Work with Existing Works	J. Sakhnini et al. [13]	**SAML-Triple**	J. Sakhnini et al. [13]	**SAML-Triple**
Feature Selection/Extraction Method	BCS	NCA	BCS	NCA
ML Algorithm Applied	SVM	(ET+AdB)	SVM	(ET+AdB)
Actual No. of Features	**34**		**137**	
No. of Features Selected/ Extracted	11 Features	**2 Components, 15 iterations, 0.001 learning rate**	94 Features	**90 Components, 5 iterations, 0.001 learning rate**
Accuracy (%)	90.69	**93.94***	88.59	**90.92***

*—represents the highest accuracy achieved through proposed approach.

Figures 11 and 12 represent the accuracy vs. parameter range graph for the IEEE 14- and 57-bus systems, respectively. The proposed approach of SAML-Triple utilizing NCA with the (ET + AdB) ML classifier outperforms the existing approach of BCS with Support Vector Machine (SVM). Based on the parameter specification range from Table 18, the proposed approach is fined-tuned to obtain a higher accuracy of 93.94% at two components, fifteen iterations, and a 0.001 learning rate compared to 90.69% with eleven features for the IEEE 14-bus system. For the IEEE 57-bus system, the proposed approach yields 90.92% accuracy at 90 components and five iterations, with a 0.001 learning rate compared to 88.59% with 94 features.

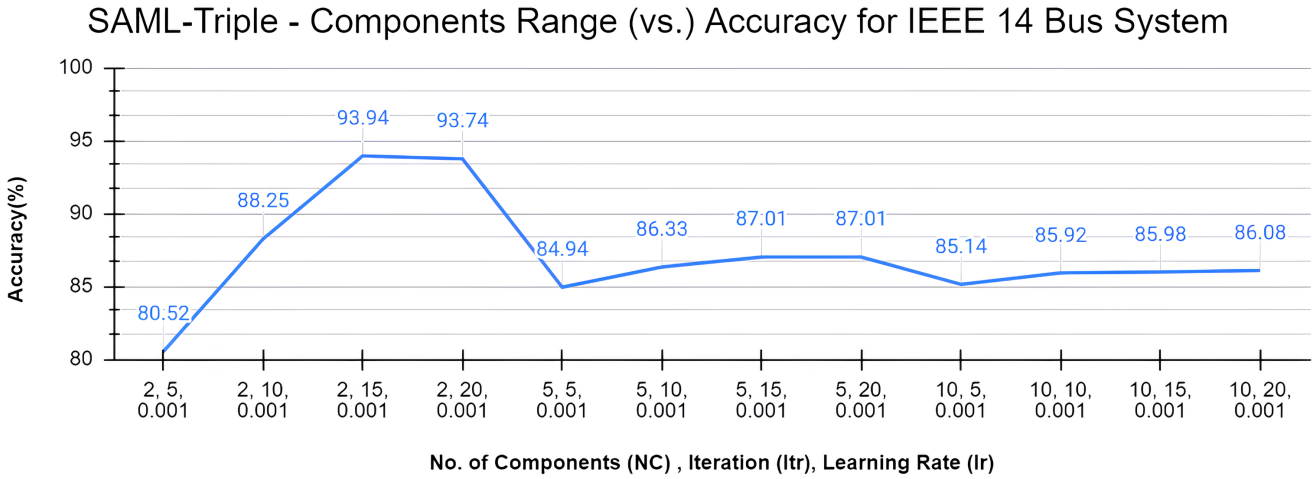


Figure 11. SAML-Triple —components range vs. accuracy for IEEE 14-bus system [38].

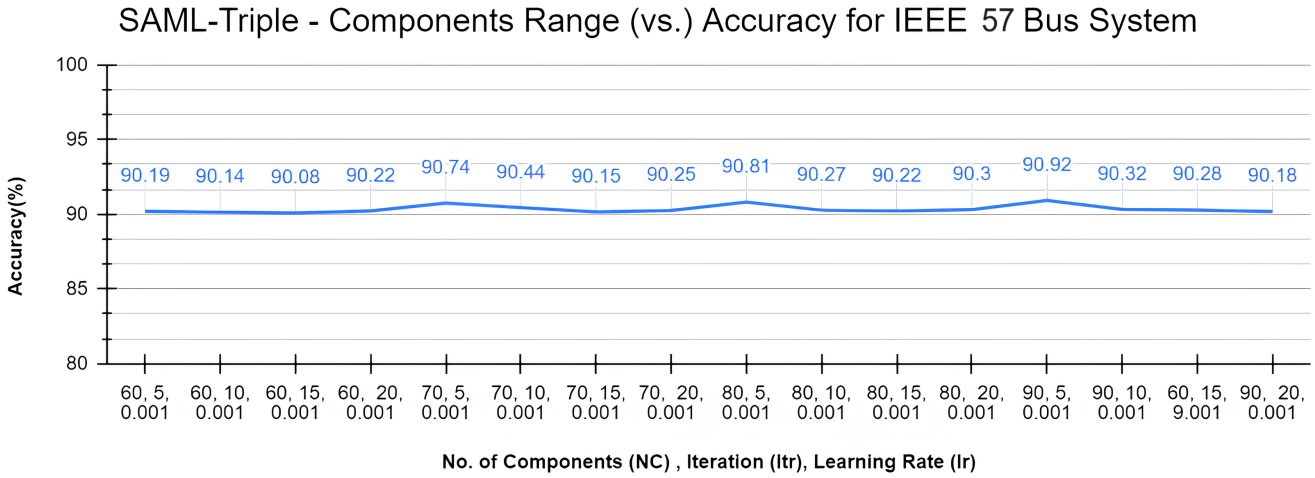


Figure 12. SAML-Triple—components range vs. accuracy for IEEE 57-bus system [38].

6.3. Overall Summary of the Proposed Work

Table 19 stands for the overall summary of the **SAML-Triple** approach with various performance metrics. In the SAML-Triple approach, comparatively higher accuracy values of 97.51% and 98.25% were obtained with INFAZ and INFAD, respectively. The FNR (missing rate) values of 2.49% and 1.75% and FPR (false alarm) values of 1.24% and 0.88% were obtained low with INFAZ and INFAD, respectively. The testing time of the IDS was 0.23 ms to detect an attack that is less than 8.3 ms between samples, by which the system admin is alerted early to activate the prevention system. Hence, the SAML-Triple approach of NCA with the (ET + AdB) ML classifier outperforms better in the discrimination of cyberattacks in triple class (No Events/Natural Events/Attack Events). Based on the comparison between the results of the existing approaches and our proposed work of SAML-Triple (INFAZ), it is concluded that SAML-Triple (INFAZ) alone can conduct this triple-class event discrimination with robustness.

Table 19. Overall summary of SAML-Triple approach with performance metrics.

Dataset Used	Proposed Work		Accuracy (%) (Detection Rate)	FNR (%) (Missing Rate)	FPR (%) (False Alarm)	Response Time (ms)
						A System [36] with 120 Samples/s Records
3-bus/2-line transmission system (Triple Class) [34]	**SAML—Triple** NCA with (ET + AdB) classifier	INFAZ	**97.51**	2.49	1.24	0.23
		INFAD	**98.25**	1.75	0.88	0.22

The robustness of the proposed SAML-Triple approach was tested for generalizability and scalability with the IEEE 14-bus and 57-bus system datasets of the FDI attacks [38]. The proposed approach outperformed with 93.94% and 90.92% for the IEEE 14-bus and 57-bus systems, respectively, compared to the existing approach with 90.69% and 88.59% accuracy.

For both the ICS Cyber Attack Power System Triple-Class Dataset and the IEEE 14- and 57-bus system datasets of the FDI attacks, the accuracy metric results were significantly impacted by the NCA parameters of the ***N-components*** and ***Iterations***. Tables 8 and 18 demonstrate that the parameters of the N-components and iterations have a significant impact on the accuracy performance metrics.

7. Conclusions and Future Works

In the mission-critical infrastructure of a smart grid, the **proposed approach** of **SAML-Triple (INFAZ)** addresses the specific problem of cyberattack discrimination from power system disturbances with a reduced missing rate and decreased response time. This paper proposes a novel mechanism of the statistical approach with Neighborhood Component Analysis as a feature extraction technique by optimal hyperparameterized tuning with the (ET + AdB) ML classifier. In the SAML-Triple approach, three events—No Events, Natural Events, and Attack Events—were discriminated with the highest accuracy of 97.51% and 98.25% by preprocessing with INFAZ and INFAD, respectively, compared to the existing approaches. The overall summary section provides insights into other performance metrics such as FNR, FPR, and response time with better results. Several test cases were executed to test the robustness of the model, which achieved more than 95%. Thus, the **SAML-Triple (INFAZ)** approach performs as a robust Anomaly-based IDS with a low missing rate of 2.49%, lower response time of 0.23 ms, decreased false alarm rate of 1.24%, and high detection accuracy of 97.51%. Our proposed novel approach addresses the privacy and access control violations of cyberattacks in the smart grid infrastructure to minimize the processing downtime, large-scale load loss, blackouts, and cascading failures. The robustness of the proposed model was evaluated with the IEEE 14-bus and 57-bus system datasets of FDI attacks for generalization and scalability, and accuracy values of 93.94% and 90.92%, respectively, were achieved.

In future work, the proposed approach will be extended to find the attack-specific location and the type of attack established from the attacker's end in the Multiclass dataset available from the same data source. The proposed approach can also be extended for scalability with other IEEE 'N' bus systems. Since data records are treated statistically with the proposed approach, it can be extended to other cyber–physical systems for anomaly detection. This proposed work can be extended to any WAMS Testbed as a smart grid by including a few more attacks from generator-side faults and insider attacks. The potential of insider attacks has not been investigated much in the context of a smart grid, and future work could focus on defining such attacks and mitigating them.

Author Contributions: Conceptualization, A.N.V. and B.S.; methodology, N.M.; software, N.M.; validation, A.N.V., B.S.P., B.S. and M.J.H.; formal analysis, N.M.; investigation, N.M.; data curation, N.M.; writing—original draft preparation, N.M.; writing—review and editing, A.N.V., B.S.P., B.S. and

M.J.H.; visualization, N.M.; supervision, A.N.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported under the Amrita Ph.D. University Scholarship for Teaching Assistant (AMRITA/CBE/CPGP/02/2024/0105).

Data Availability Statement: The Triple Class Power System Cyber Attack datasets used in this study are available from http://www.ece.uah.edu/~thm0009/icsdatasets/triple.7z (accessed on 10 February 2023) at https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets (accessed on 10 February 2023) [34]. The IEEE 14- and 57-bus systems of FDI attack datasets used in this study are available from [HeuristicFS] at https://github.com/jsakhnin/HeuristicFS/tree/master/input (accessed on 11 December 2023) [13,38]. These data were derived from the above resources available in the public domain.

Acknowledgments: The authors are pleased to acknowledge the infrastructure support provided through the Internal—Amrita Seed Grant Fund Program (File No: ASG2022051) from the School of Computing, Department of CSE, and Department of EEE, Amrita Vishwa Vidyapeetham, Coimbatore, India to complete this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SAML	Statistical Approach with Machine Learning
NCA	Neighborhood Components Analysis
PCA	Principal Component Analysis
INFAZ	INFinity—Attack Events Records as Zero
INFAD	INFinity—Attack Events Records by Dropping
SMOTE	Synthetic Minority Oversampling Technique
ET + AdB	ExtraTrees with AdaBoost classifier
FNR	False Negative Rate
FPR	False Positive Rate
SCADA	Supervisory Control and Data Acquisition
PMU	Phasor Measurement Unit
WAMS	Wide Area Measurement Systems
CPPS	Cyber–Physical Power Systems
ICS	Industrial Control Systems
SLG	Single Line-to-Ground Fault
IDS	Intrusion Detection System
FDI	False Data Injection Attack

References

1. Yohanandhan, R.V.; Elavarasan, R.M.; Manoharan, P.; Mihet-Popa, L. Cyber-Physical Power System (CPPS): A Review on Modeling, Simulation, and Analysis with Cyber Security Applications. *IEEE Access* **2020**, *8*, 151019–151064. [CrossRef]
2. Gunduz, M.Z.; Das, R. Cyber-Security on Smart Grid: Threats and Potential Solutions. *Comput. Netw.* **2020**, *169*, 107094. [CrossRef]
3. Kimani, K.; Oduol, V.; Langat, K. Cyber Security Challenges for IoT-Based Smart Grid Networks. *Int. J. Crit. Infrastruct. Prot.* **2019**, *25*, 36–49. [CrossRef]
4. Hemsley, K.E; Fisher, E. History of Industrial Control System Cyber Incidents. 2018. Available online: https://www.osti.gov/biblio/1505628/ (accessed on 10 February 2023).
5. Gupta, P.K.; Narayanan Babu, S.S.; Mohandas Sheeladevi, A.; Pampana, V. Why Dealing with Electrical Faults for Smart Microgrid is not Enough? In Proceedings of the Science and Technologies for Smart Cities, Virtual, 2–4 December 2021; pp. 55–74. [Crossref]
6. Zhang, C.; Lu, Z.; Zhu, Z.; Shi, Z.; Xu, X.; Yan, Z. Demonstration Project and State Estimation Application in PMU-Based Distribution Network. In Proceedings of the 2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2), Wuhan, China, 30 October–1 November 2020; pp. 1000–1004. [Crossref]
7. Zhang, J.E.; Wu, D.; Boulet, B. Time Series Anomaly Detection for Smart Grids: A Survey. In Proceedings of the 2021 IEEE Electrical Power and Energy Conference (EPEC), Toronto, ON, Canada, 22–31 October 2021; pp. 125–130. [Crossref]
8. Worldwide, Capgemini. Reinventing Cybersecurity with Artificial Intelligence: The New Frontier in Digital Security. 2020. Available online: https://www.capgemini.com/wp-content/uploads/2019/07/AI-in-Cybersecurity_Report_20190711_V06.pdf (accessed on 10 February 2023).

9. Chawla, A.; Panigrahi, B.K.; Bhalja, B.R. Deep-Learning-Based Denial-of-Service Resilient Framework for Wide Area Situational Awareness of Power Systems. *IEEE Trans. Ind. Inform.* **2023**, *19*, 9204–9216. [CrossRef]
10. Singh, V.K.; Govindarasu, M. A Cyber-Physical Anomaly Detection for Wide-Area Protection Using Machine Learning. *IEEE Trans. Smart Grid* **2021**, *12*, 3514–3526. [CrossRef]
11. Amin, B.M.R.; Hossain, M.J.; Anwar, A.; Zaman, S. Cyber Attacks and Faults Discrimination in Intelligent Electronic Device-Based Energy Management Systems. *Electronics* **2021**, *10*, 650. [CrossRef]
12. Kumar, A.; Saxena, N.; Jung, S.; Choi, B.J. Improving Detection of False Data Injection Attacks Using Machine Learning with Feature Selection and Oversampling. *Energies* **2021**, *15*, 212. [CrossRef]
13. Sakhnini, J.; Karimipour, H.; Dehghantanha, A. Smart Grid Cyber Attacks Detection Using Supervised Learning and Heuristic Feature Selection. In Proceedings of the 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–14 August 2019; pp. 108–112. [Crossref]
14. Faramondi, L.; Flammini, F.; Guarino, S; Setola, R. Evaluating Machine Learning Approaches for Cyber and Physical Anomalies in SCADA Systems. In Proceedings of the 2023 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy, 31 July–2 August 2023; pp. 412–417. [Crossref]
15. Li, X.J.; Ma, M.; Sun, Y. An Adaptive Deep Learning Neural Network Model to Enhance Machine-Learning-Based Classifiers for Intrusion Detection in Smart Grids. *Algorithms* **2023**, *16*, 288. [CrossRef]
16. AlHaddad, U.; Basuhail, A.; Khemakhem, M.; Eassa, F.E.; Jambi, K. Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks. *Sensors* **2023**, *23*, 7464. [CrossRef]
17. Unsal, D.B.; Ustun, T.S.; Hussain, S.M.S.; Onen, A. Enhancing Cybersecurity in Smart Grids: False Data Injection and Its Mitigation. *Energies* **2021**, *14*, 2657. [CrossRef]
18. Radoglou-Grammatikis, P.I.; Sarigiannidis, P.G. Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems. *IEEE Access* **2019**, *7*, 46595–46620. [CrossRef]
19. Ozkan-Okay, M.; Samet, R.; Aslan, O.; Gupta, D. A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access* **2021**, *9*, 157727–157760. [CrossRef]
20. Pan, S.; Morris, T.; Adhikari, U. Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems. *IEEE Trans. Smart Grid* **2015**, *6*, 3104–3113. [CrossRef]
21. Chawla, A.; Singh, A.; Agrawal, P.; Panigrahi, B.K.; Bhalja, B.R.; Paul, K. Denial-of-Service Attacks Pre-Emptive and Detection Framework for Synchrophasor Based Wide Area Protection Applications. *IEEE Syst. J.* **2022**, *16*, 1570–1581. [CrossRef]
22. Upadhyay, D.; Manero, J.; Zaman, M.; Sampalli, S. Gradient Boosting Feature Selection with Machine Learning Classifiers for Intrusion Detection on Power Grids. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1104–1116. [CrossRef]
23. Upadhyay, D.; Manero, J.; Zaman, M.; Sampalli, S. Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model with Majority Vote Ensemble Algorithm. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2559–2574. [CrossRef]
24. Hu, C.; Yan, J.; Wang, C. Robust Feature Extraction and Ensemble Classification Against Cyber-Physical Attacks in the Smart Grid. In Proceedings of the 2019 IEEE Electrical Power and Energy Conference (EPEC), Montreal, QC, Canada, 16–18 October 2019; pp. 1–6. [Crossref]
25. Hu, C.; Yan, J.; Liu, X. Adaptive Feature Boosting of Multi-Sourced Deep Autoencoders for Smart Grid Intrusion Detection. In Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2–6 August 2020; pp. 1–5. [Crossref]
26. Gumaei, A.; Hassan, M.M.; Huda, S.; Hassan, Md.R.; Camacho, D.; Del Ser, J.; Fortino, G. A Robust Cyberattack Detection Approach Using Optimal Features of SCADA Power Systems in Smart Grids. *Appl. Soft Comput.* **2020**, *96*, 106658. [CrossRef]
27. Ankitdeshpandey; Karthi, R. Development of Intrusion Detection System Using Deep Learning for Classifying Attacks in Power Systems. In Proceedings of the Soft Computing: Theories and Applications, Singapore, 30 June 2020; pp. 755–766. [Crossref]
28. Hink, R.C.B.; Beaver, J.M.; Buckner, M.A.; Morris, T.; Adhikari, U.; Pan, S. Machine learning for power system disturbance and cyber-attack discrimination. In Proceedings of the 2014 7th International Symposium on Resilient Control Systems (ISRCS), Denver, CO, USA, 9–21 August 2014; pp. 1–8. [Crossref]
29. Agrawal, A.; Sazos, M.; Al Durra, A; Maniatakos, M. Towards Robust Power Grid Attack Protection using LightGBM with Concept Drift Detection and Retraining. In Proceedings of the Proceedings of the 2020 Joint Workshop on CPS & IoT Security and Privacy, Virtual Event, 9 November 2020; pp. 31–36. [Crossref]
30. Sunku Mohan, V.; Sankaran, S. Intelligent Approach for Analysis and Diagnosis of Attack, Fault and Load Variation in SCADA Systems: A Power System Application. In Proceedings of the Intelligent Data Analytics for Power and Energy Systems, Singapore, 17 February 2022; pp. 1–28. [Crossref]
31. Bitirgen, K.; Filik, Ü.B. A Hybrid Deep Learning Model for Discrimination of Physical Disturbance and Cyber-Attack Detection in Smart Grid. *Int. J. Crit. Infrastruct. Prot.* **2023**, *40*, 100582. [CrossRef]
32. Yang, C.; Xia, Y. Interval Pareto Front-Based Multi-Objective Robust Optimization for Sensor Placement in Structural Modal Identification. *Reliab. Eng. Syst. Saf.* **2024**, *242*, 109703. [CrossRef]
33. Gao, J.; Chai, S.; Zhang, B.; Xia, Y. Research on Network Intrusion Detection Based on Incremental Extreme Learning Machine and Adaptive Principal Component Analysis. *Energies* **2019**, *12*, 1223. [CrossRef]

34. Hink, R.C.B.; Beaver, J.M.; Buckner, M.A.; Morris, T.; Adhikari, U; Pan, S. Industrial Control System (ICS) Cyber Attack Datasets Used in the Experimentation. 2014. Available online: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets (accessed on 10 February 2023).
35. Balan, A.; Srujan, T.L.; Manitha, P.V.; Deepa, K. Detection and Analysis of Faults in Transformer using Machine Learning. In Proceedings of the 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 5–7 January 2023; pp. 477–482. [Crossref]
36. Pan, S.; Morris, T.; Adhikari, U. Classification of Disturbances and Cyber-Attacks in Power Systems Using Heterogeneous Time-Synchronized Data. *IEEE Trans. Ind. Inform.* **2015**, *11*, 650–662. [CrossRef]
37. Goldberger, J.; Hinton, G.E.; Roweis, S; Salakhutdinov, R.R. Neighbourhood components analysis. In Proceedings of the 17th International Conference on Advances in Neural Information Processing Systems (NIPS 2004), Vancouver, BC, Canada, 13–18 December 2004; pp. 513–520. [Crossref]
38. Sakhnini, J. HeuristicFS. 2020. Available online: https://github.com/jsakhnin/HeuristicFS (accessed on 11 December 2023).