CoreID JOURNAL

Vol. 4 No. 2, July 2026 (78-90)
ISSN: 2987-6990
DOI: https://doi.org/10.60005/coreid.v4i2.176

PhishTect: A Hybrid SMOTEN–FT-Transformer–PSO Framework for Enhanced Phishing Website Detection on Imbalanced Data

M. Hizbul Wathan[1], Muhammad Kalam Sabili[2]
[1] Department of Informatics and Business, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, Indonesia

Article Info

Article history:

Received June 05, 2026
Revised August 05, 2026
Accepted August 24, 2026

Keywords:

FT-Transformer
Imbalanced Data Classification
Particle Swarm Optimization
Phishing Website Detection
SMOTEN

ABSTRACT

Phishing websites remain a major cybersecurity threat, while conventional blacklist-based detection systems often fail to identify newly emerging and zero-day attacks. In addition, phishing datasets are frequently imbalanced, causing machine learning models to exhibit poor minority-class detection performance. To address these challenges, this study proposes PhishTect, a hybrid phishing detection framework that integrates Synthetic Minority Oversampling for Nominal Data (SMOTEN), Feature Tokenizer Transformer (FT-Transformer), and Particle Swarm Optimization (PSO). Unlike existing approaches that typically focus on balancing, deep learning, or optimization techniques separately, the proposed framework combines these components within a unified architecture. SMOTEN is employed to balance categorical phishing data, FT-Transformer learns contextual representations from tabular features, and PSO optimizes model hyperparameters to improve predictive capability. Experiments conducted on the PhishTank dataset evaluated three scenarios: baseline FT-Transformer, FT-Transformer with SMOTEN, and the proposed SMOTEN–FT-Transformer–PSO framework. The proposed model achieved the best performance, obtaining 96.62% accuracy, 0.9696 F1-score, 0.9963 ROC-AUC, and 0.997 PR-AUC. The results demonstrate that integrating oversampling, Transformer-based feature learning, and swarm intelligence optimization significantly improves phishing detection effectiveness, robustness, and generalization on imbalanced cybersecurity datasets.

Corresponding Author:

M. Hizbul Wathan
Department of Informatics and Business, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, Indonesia
Email:mhizbul@polman-babel.ac.id

1. INTRODUCTION

Phishing has become one of the most damaging cyber threats since it was first reported in 1995. According to the Anti-Phishing Working Group (APWG), more than 1 million phishing incidents were recorded in the first quarter of 2025 alone, the highest number since late 2023 [1]. The IBM X-Force report also shows that although the proportion of traditional phishing as an initial attack vector decreased from 46 percent in 2022 to around 25 percent in 2024, attackers are increasingly exploiting valid credentials to gain initial access [2]. The impact of phishing is multidimensional, including global financial losses of billions of dollars each year, operational disruption to organizations, and reduced public trust in the digital ecosystem [3].

This highlights that phishing detection is not only a technical challenge but also a matter of sustaining the digital economy and protecting society at large.

Traditional blocklist-based approaches have proven inadequate because they cannot detect novel zero-day attacks [4]. Machine learning (ML) and deep learning (DL) have therefore been widely adopted for their ability to learn complex patterns in phishing data [5]. However, studies of meta-learning in other domains show that class imbalance and limited generalization remain significant obstacles [6]. A comprehensive review of deep learning for phishing detection emphasizes that the issues of imbalanced data, parameter optimization, and the need for more transparent models are still important research gaps [7]. These challenges not only reduce detection accuracy but also leave security vulnerabilities that can be exploited by attackers.

Many researchers have focused on developing more adaptive classification models using ML and DL. Architectures based on convolutional neural networks (CNN), recurrent neural networks (RNN), bidirectional RNN (BRNN), and attention mechanisms have demonstrated effectiveness in extracting complex patterns from phishing URLs and have achieved high detection performance [8]. To reduce the bias caused by skewed class distributions, DeepSMOTE has been proposed as an oversampling strategy in the representation space, thereby improving the representation of minority classes [9]. Other efforts combine CNN with support vector machines (SVM) optimized through metaheuristics, which take advantage of automatic feature extraction while improving classification accuracy [10]. In email phishing detection, algorithms such as SVM, Random Forest, and XGBoost have been combined with feature selection techniques to achieve competitive performance [11], while comparative studies using Random Forest, SVM, and neural networks have provided baselines for evaluating newer methods [12]. More recently, approaches that integrate diffusion models with multi-head self-attention have shown the potential of combining generative and discriminative architectures to improve detection on imbalanced datasets[13].

Class imbalance remains a major challenge in phishing detection because models tend to be biased toward the majority class. Balancing strategies have included combining oversampling with data cleaning techniques, such as ROSE with Tomek Links [14], as well as developing probabilistic variants of SMOTE that are more robust to outliers [15]. In other domains, combining BiLSTM with SMOTE and ADASYN has improved the accuracy of extreme-weather prediction [16], while multi-scale semantic fusion architectures have achieved F1-scores above 0.98 for phishing detection [17]. In addition, XGBoost optimized with metaheuristics such as the Bat Algorithm has achieved better adaptive accuracy [18], and TTVAE-based generative approaches have provided a more flexible alternative for synthesizing tabular data compared to SMOTE [19]. These findings indicate that integrating balancing techniques with adaptive optimization is a promising direction for improving the reliability of phishing detection in imbalanced data.

To address these challenges, this study proposes a phishing detection framework based on SMOTEN combined with FT-Transformer and Particle Swarm Optimization (PSO). Rather than functioning as separate techniques, these three components are designed to complement each other. SMOTEN balances the class distribution and reduces bias toward the majority class. The Feature Tokenizer-Transformer provides deep representational capacity tailored for tabular phishing data. PSO serves as a metaheuristic optimizer that adjusts the hyperparameters so that the model can operate efficiently [20]. Transformer-based approaches have already been shown to be effective in detecting URL-based phishing [21], and PSO has previously been successfully applied to feature selection in related domains [22].

The contributions of this study are threefold. First, it presents a systematic evaluation across three experimental scenarios: the baseline FT-Transformer, the combination of SMOTEN and FT-Transformer, and the hybrid combination of SMOTEN, FT-Transformer, and PSO. This evaluation provides a comprehensive view of how balancing and optimization influence phishing detection performance. Second, the integration of balancing, a tabular Transformer architecture, and metaheuristic optimization creates a hybrid framework that is adaptive to imbalanced distributions and remains computationally efficient. Third, this study introduces a novel synergy of balancing techniques, deep representation learning, and metaheuristic optimization, a combination that has rarely been explored in phishing detection research. This integration is expected to improve the reliability of digital security systems. The experiments are designed to show that combining SMOTEN and PSO with the FT-Transformer not only enhances metrics such as the F1-score and ROC-AUC compared with the baseline but also improves sensitivity to the minority class, thereby addressing the core challenges identified in earlier research.

2. METHOD

2.1 Proposed Method

The overall framework of the proposed method is illustrated in Figure 1. The design integrates three main components: (i) a preprocessing pipeline with SMOTEN for handling imbalanced categorical data, (ii) an FT-Transformer architecture tailored for tabular feature representation, and (iii) hyperparameter tuning through Particle Swarm Optimization (PSO). As shown in the figure, the raw dataset is first subjected to feature cleansing, categorical encoding, and stratified data splitting. To address the skewed class distribution, SMOTEN is applied for generating synthetic minority-class samples in a manner consistent with categorical feature distributions. The balanced data are then fed into the Feature Tokenizer module, which projects each categorical attribute into a dense embedding space suitable for Transformer-based processing.

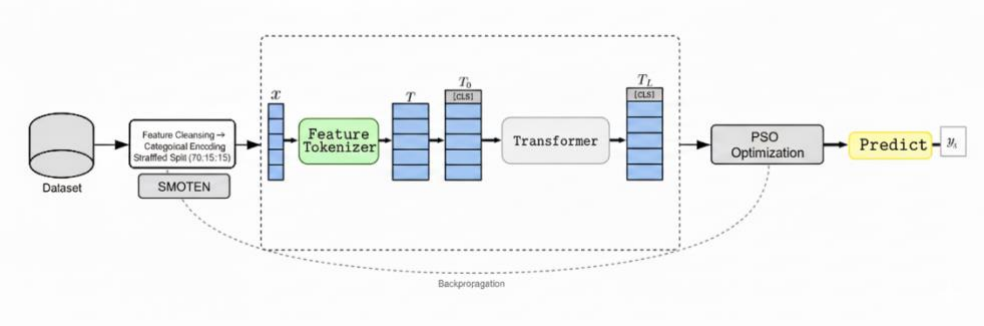

Figure 1. Proposed Method

Within the Transformer block, the embedded features interact through multi-head self-attention, enabling the model to capture complex dependencies across attributes. The final representation from the [CLS] token is then optimized using PSO, which adaptively tunes hyperparameters to improve predictive performance. The prediction layer produces the final output, corresponding to whether a given URL is phishing or legitimate. This integration of SMOTEN, FT-Transformer, and PSO provides a unified framework that addresses key challenges in phishing detection: (i) imbalanced data distribution, (ii) effective feature representation, and (iii) hyperparameter optimization for robust generalization.

2.2 Dataset

The dataset used in this study came from the UCI Machine Learning Repository, specifically the Phishing Websites dataset.This dataset was created using information about phishing websites collected from PhishTank, which is a public place where people share and check reports about real phishing websites [23], [24]. The dataset includes 11,055 website examples, with 4,898 being phishing sites and 6,157 being regular websites.Each website is described using 30 features that were created by humans and show details about the URL and other website traits. The dataset is publicly available and has no missing values, so the preprocessing step mainly involved checking the data for accuracy, converting the features into a suitable format, and getting the data ready for training the model.

Following the method used by Mohammad et al.[24], the dataset was handled using automatic feature extraction and rule-based feature generation. This process captured both URL-based features like URL length, use of IP addresses, presence of special characters, and whether HTTPS is valid, as well as content-based features taken from HTML and JavaScript, such as requests for external resources, form actions, and DNS-related settings. These features, which have been shown to work well in detecting phishing websites, are listed and grouped in Table 1. The set of features created is used for a task that sorts websites into two groups, with phishing sites marked as −1 and real websites marked as 1.

Table 1. Dataset's sample

id	having_IP _Address	URL_ Length	Shortening_ Service	having_At_ Symbol	...	Page_ Rank	Google _Index	Links_pointing _to_page	Statistical _report	Result
0	1	-1	1	1	...	-1	1	1	-1	-1
1	2	1	1	1	...	-1	1	1	1	-1
2	3	1	0	1	...	-1	1	0	-1	-1
…	…	…	…	…	…	…	…	…	…	…

2.3 Pre-Processing

The pre-processing stage was carried out to ensure that the data were ready to be processed by the machine learning model [25]. The process began by removing the id column, which served only as a unique

identifier and had no predictive value [26]. The classification target was defined as the Result column, which represents the phishing and legitimate classes in binary format, while the remaining thirty discrete columns were retained as predictive features after excluding id and Result [27]. All features were confirmed to be categorical variables, and therefore label encoding was applied to convert each category into an integer-based numerical representation suitable for the embedding mechanism of the FT-Transformer model. As there were no numerical features, additional normalization or scaling was not required.

The dataset was then split into three subsets with a ratio of 70% for training, 15% for validation, and 15% for testing, using a stratified splitting strategy to preserve the class proportion in each subset [28], [29]. This procedure ensured that the data used for training and evaluating the model maintained a consistent and representative distribution.

2.4 Imbalance Handling

Although the data were split using a stratified strategy, class imbalance remained a challenge because the number of phishing sites was smaller than that of legitimate sites. To address this issue, this study employed the Synthetic Minority Over-sampling Technique for Nominal (SMOTEN) [27]. SMOTEN is an extension of SMOTE, specifically designed to handle categorical features [28]. In general, the SMOTE algorithm generates synthetic samples by selecting the nearest neighbors from the minority class and performing linear interpolation, which is mathematically expressed as:

$$\tilde{x} = x_i + \delta \times \left(x_{\{zi\}} - x_i\right),\ \delta \sim U(0,1) \quad (1)$$

Where $x_{\{zi\}}$ denotes the feature vector of a minority-class instance, $x_{\{zj\}}$represents one of its nearest neighbors selected from the same class, and δ is a random value drawn from the uniform distribution (U(0,1)). In SMOTEN, linear interpolation is applied only to numerical features. For categorical features, new values are generated using majority voting among the nearest neighbors to preserve the original data distribution [27], [28].This approach enables the creation of representative synthetic samples without introducing excessive noise, as recommended in recent studies addressing data imbalance in both healthcare [25] and cybersecurity domains [29]. By balancing the class distribution, machine learning models such as XGBoost can better capture patterns in the minority class, thereby improving accuracy and recall in phishing detection.

2.5 Hyperparameter Optimization

To optimize the performance of the FT-Transformer model, this study adopts Particle Swarm Optimization (PSO) as an adaptive hyperparameter search algorithm. PSO is a population-based metaheuristic inspired by the collective behavior of bird flocks and fish schools in searching for food sources [25]. In PSO, each candidate solution is represented as a particle with a position $x_{i(t)}$ and a velocity $v_{i(t)}$ in the search space, where the particle's position corresponds to a specific combination of hyperparameter values being evaluated. The evolution of a particle's position is carried out by updating its velocity using the following equations:

$$v_{i(t+1)} = w \cdot v_{i(t)} + c_1 r_1 \left(p_i^{\{best\}} - x_{i(t)}\right) + c_2 r_2 \left(g^{\{best\}} - x_{i(t)}\right) \quad (2)$$

$$x_{i(t+1)} = x_{i(t)} + v_{i(t+1)} \quad (3)$$

where w is the inertia weight that controls the balance between exploration and exploitation, c_1 and c_2 are the cognitive and social coefficients, respectively, and r_1, r_2 are random numbers uniformly distributed in [0,1]. The term $p_i^{\{best\}}$ represents the best position ever achieved by particle i, while $g^{\{best\}}$ denotes the best position found by the entire swarm.

This approach has been shown to be effective in accelerating the convergence of neural network training while reducing prediction errors in medical domains, such as for the diagnosis of chronic kidney disease[25]. In cervical cancer prediction, PSO has also been used to optimize the configuration of FT-Transformer, resulting in more effective tabular representations [26]. Furthermore, in the field of cybersecurity, the use of metaheuristics for hyperparameter optimization has been proven to enhance phishing detection performance, as demonstrated by the Bat Algorithm integrated with XGBoost [18]. Therefore, the integration of PSO in this study is expected to provide a more adaptive parameter search mechanism, improve the model's generalization capability, and strengthen the accuracy of phishing detection in imbalanced data. Before the

tuning process, however, a set of default parameters was initialized for both the FT-Transformer and the PSO algorithm, as summarized in Table 2. These values serve as the starting point for the optimization process and define the search boundaries explored during training.

Table 2. PSO's Parameters

Parameter	Default Value	Notes
n_particles	6	Number of particles in swarm
n_iter	6	Iterations per tuning run
w	0.7	Inertia weight
c1	1.4	Cognitive coefficient (individual best influence)
c2	1.4	Social coefficient (global best influence)
seed	42	Random seed for reproducibility

The PSO parameters were set with a smaller swarm size (`n_particles = 6`) and a limited number of iterations (`n_iter = 6`) because PSO was used only for hyperparameter optimization in a low-dimensional space. Because each time we check a particle, we have to fully train and test the FT-Transformer model, making the computation more expensive if we use a larger group of particles or more rounds of checking. So, a smaller PSO setup was used to keep things efficient while still checking out all the possible hyperparameter options.

2.6 Model Architecture (Feature Tokenizer Transformer)

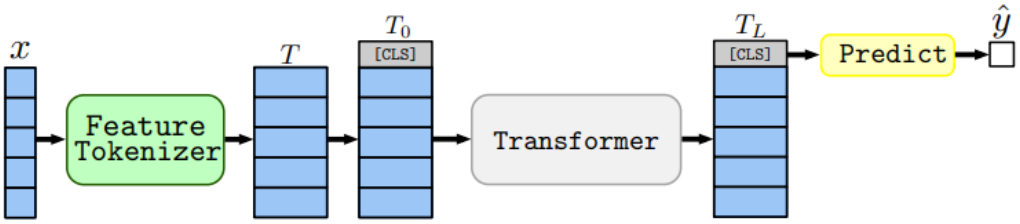

Figure 2. FT – Transformer's Framework

FT-Transformer (Feature Tokenizer Transformer) is an adaptation of the Transformer architecture specifically designed for tabular data. Unlike text or image data, which have sequential or spatial structures, tabular data consist of both numerical and categorical features with diverse distributions. To address this, FT-Transformer treats each feature as a token and projects it into a fixed-dimensional embedding space so that it can be modeled through the self-attention mechanism.

As illustrated in Figure 2, the architecture of FT-Transformer consists of three main stages. The first stage is feature tokenization, in which each feature is transformed into a vector representation. Numerical features are typically projected through a linear layer, while categorical features are represented using embeddings. The second stage enriches the feature embeddings with feature encodings that differentiate the positions or identities of columns. In the third stage, the set of tokens is processed through Transformer blocks composed of multi-head self-attention (MHA), residual connections, normalization layers, and feed-forward networks. This structure enables the model to capture complex interactions among features that are often difficult for traditional multilayer perceptrons (MLPs) to learn [23].

The self-attention mechanism in the FT-Transformer follows the formulation of scaled dot-product attention. Given an input embedding matrix $X \in R^{n \times d}$, where is the number of features and is the embedding dimension, the query , key , and value matrices are obtained through linear projections. The attention function is defined as:

$$Attention(Q,K,V) = softmax\left(\{QK^{\{T\}}\}\left\{\sqrt{\{d_k\}}\right\}\right)V \quad (4)$$

In the multi-head architecture, several attention heads are computed in parallel to capture diverse representations of feature interactions. The outputs of all heads are then concatenated and projected back through a weight matrix W_O:

$$MHA(X) = Concat(head_1, \dots, head_h)W_O \quad (5)$$

$$head_i = Attention\left(XW_Q^{(i)}, XW_K^{(i)}, XW_V^{(i)}\right) \quad (6)$$

Several studies have demonstrated the advantages of FT-Transformer for tabular data. A comprehensive review of deep learning models for tabular tasks found that FT-Transformer consistently performs competitively with gradient boosting–based models. Furthermore, the development of fault-tolerant attention within FT-Transformer has been proposed to improve reliability in large-scale computing with minimal overhead[30].

In the field of network security, FT-Transformer has outperformed CNNs, RNNs, and autoencoders in IoT intrusion detection by leveraging the interactions among network traffic features [31]. In medical applications, FT-Transformer has been combined with convolutional modules and LSTMs for cervical cancer prediction, achieving high accuracy while supporting interpretability through SHAP and LIME [26].

Therefore, the FT-Transformer employed in this study is grounded in the fundamental principles of the Transformer architecture and reinforced by empirical evidence from diverse domains, including fault-tolerant computing, IoT network security [31], and tabular data-based medical prediction [26]. To provide clarity regarding the implementation, the main hyperparameters of the FT-Transformer are listed in Table 3s. These values represent the baseline configuration before optimization.

Table 3. FT-Transdormer's Parameters

Parameter	Default Value	Notes
emb_dim	32	Embedding dimension per feature token
d_model	128	Transformer hidden dimension
n_heads	4	Number of attention heads
n_blocks	3	Number of Transformer encoder layers
dropout	0.1	Dropout rate in encoder and head
n_classes	2	Binary classification (phishing vs legitimate)

As shown in Table 3, the parameters primarily control the representational capacity and regularization of the FT-Transformer. For example, increasing the embedding dimension (emb_dim) allows richer feature representations, while the number of heads (n_heads) determines how many parallel feature interactions are modeled. These defaults provided a balanced starting point and were later optimized through the PSO process.

2.7 Evaluation

The model's performance was evaluated using six primary metrics: accuracy, precision, recall, F1-score, the area under the receiver operating characteristic curve (ROC-AUC), and the area under the precision–recall curve (PR-AUC). The selection of these metrics follows common practice in recent literature, which emphasizes comprehensive evaluation, particularly when dealing with imbalanced data [6],[28], [32].

Accuracy measures the proportion of correct predictions over the total number of test instances and is defined as:

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) \quad (7)$$

Although simple, accuracy often does not adequately represent performance on imbalanced datasets. Therefore, precision and recall are also used. Precision measures the proportion of correct positive predictions:

$$Precision = \left(\frac{TP}{TP+FP}\right) \quad (8)$$

Meanwhile, recall measures the model's ability to identify all positive cases:

$$Recall = \left(\frac{TP}{TP+FN}\right) \quad (9)$$

These two metrics often exhibit a trade-off; therefore, the F1-score is used as the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\Big) \quad (10)$$

In addition to these confusion-matrix-based metrics, this study also employs ROC-AUC and PR-AUC. ROC-AUC is derived from the ROC curve, which plots the true positive rate (TPR) against the false positive rate (FPR) across various threshold values:

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN}) \tag{11}$$

The area under the ROC curve reflects the overall discriminative capability of the model. However, in datasets with a high degree of class imbalance, PR-AUC is often considered more relevant. PR-AUC is calculated from the precision–recall curve, which emphasizes the model's performance in detecting the minority class [28]. The use of these six metrics aligns with evaluation standards reported in relevant literature. For example, in cervical cancer prediction using FT-Transformer, accuracy, F1-score, and ROC-AUC were employed to assess performance on imbalanced medical data [26]. In stroke prediction with attention-based meta-learning, the combination of precision, recall, and F1-score was shown to be crucial for demonstrating the model's sensitivity.

Studies on oversampling in deep learning have highlighted PR-AUC as a primary indicator of model quality in healthcare applications [28], while comprehensive analyses of XGBoost and Random Forest have identified PR-AUC and ROC-AUC as the most informative metrics for imbalanced datasets. Finally, in phishing detection using XGBoost, accuracy and ROC-AUC have been emphasized due to the practical importance of minimizing false positives for user security. Therefore, the combined use of accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC provides a comprehensive evaluation of model performance, capturing both overall correctness and the ability to detect the minority class. This evaluation strategy is consistent with best practices reported in recent literature across healthcare, cybersecurity, and tabular machine learning [6], [18], [26], [28].

3. RESULTS AND DISCUSSION

3.1 Baseline FT-Transformer

The initial experiment employed the FT-Transformer without any data balancing techniques or hyperparameter optimization. The quantitative evaluation showed an accuracy of 95.3%, an F1-score of 0.958, a ROC-AUC of 0.993, and a PR-AUC of 0.995. The model achieved a precision of 0.97 for the positive class, while the recall remained at 0.95.

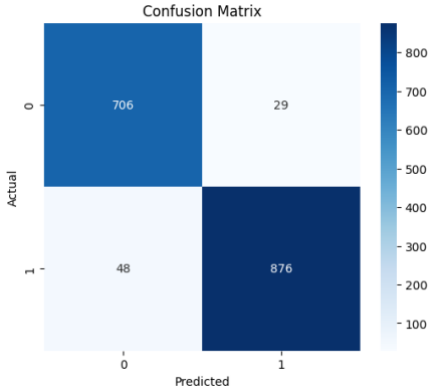

Figure 3. Baseline FT-Transformer's Confusion Matrix

The confusion matrix in Figure 3 indicates that the model was generally balanced in detecting both classes, although the number of false negatives was relatively higher than that of false positives. This suggests that the model was slightly more conservative in identifying instances of the minority class.

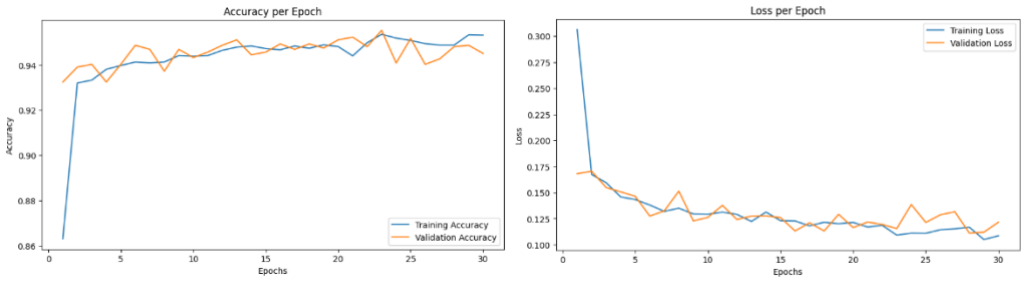

Figure 4. Baseline FT-Transformer's Acc and Loss graphs

The accuracy and loss curves in Figure 4 demonstrate stable convergence with a consistently decreasing loss, despite minor fluctuations toward the end of training. In Figure 5, the ROC approaches the top-left corner, indicating strong discriminative capability. As for the PRC curve shows an almost perfect area, although precision declines slightly at recall levels approaching 1.

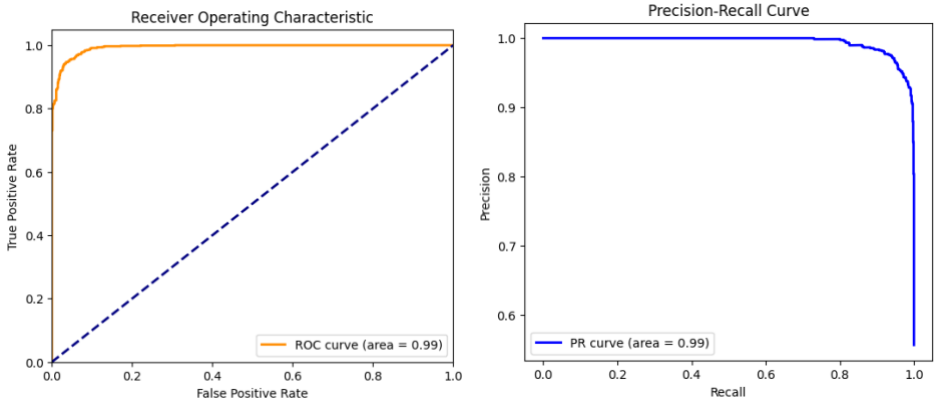

Figure 5. Baseline FT-Transformer's ROC and PRC graphs

3.2 FT-Transformer with SMOTEN

In the second experiment, the data were balanced using SMOTEN prior to training. The impact was immediately evident in the evaluation results, with accuracy increasing to 95.5% and the F1-score reaching 0.959. The ROC-AUC improved to 0.994, while the PR-AUC remained high at 0.995.

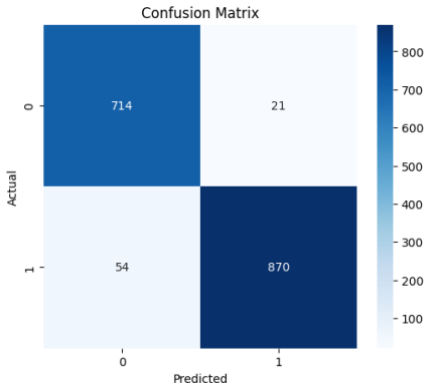

Figure 6. FT-Transformer with SMOTEN's Confusion Matrix

The confusion matrix in Figure 6 shows a noticeable reduction in false negatives compared with the baseline model. Although the recall of Class 1 changed only marginally (from 0.95 to 0.94), the application of SMOTEN resulted in a more balanced classification performance across both classes, as reflected by the improved F1-score and reduced prediction bias toward the majority class.

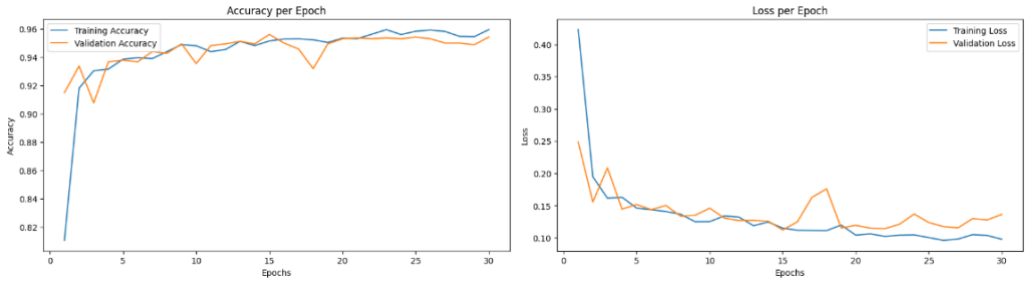

Figure 7. FT-Transformer with SMOTEN's Acc and Loss graphs

The accuracy and loss curves in Figure 7 reveal a more stable convergence pattern compared with the baseline, with a smaller gap between training and validation performance. In Figure 8, the ROC curve demonstrates an improved margin in regions of high sensitivity, while the PRC curve indicates that the model maintained consistent precision even as recall increased.

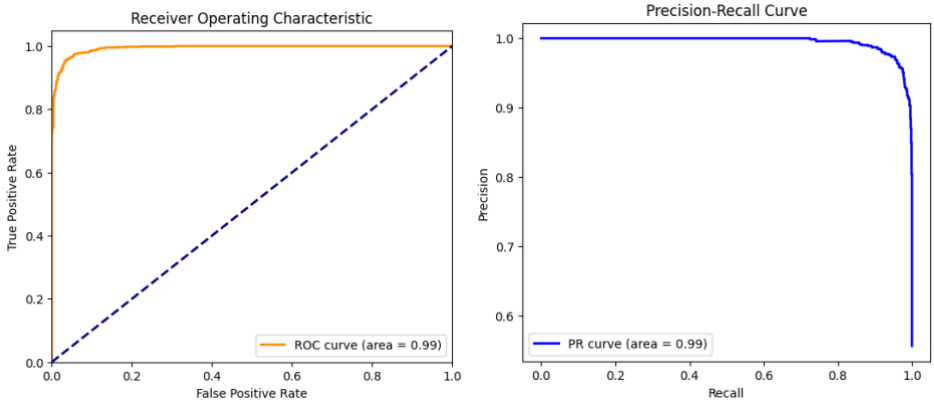

Figure 8. FT-Transformer with SMOTEN's ROC and PRC graphs

3.3 FT-Transformer with SMOTEN and PSO

In the third experiment, SMOTEN was combined with Particle Swarm Optimization (PSO) for hyperparameter tuning. The optimal configuration obtained included a learning rate of , a weight decay , an embedding dimension of 33, , three attention heads, three Transformer blocks, and a dropout rate of 0.176. This configuration achieved the highest validation F1-score of 0.952, which was subsequently confirmed on the test set. The evaluation results indicate a steady improvement, with accuracy reaching approximately 96.6%, the highest F1-score among all experiments, and ROC-AUC and PR-AUC consistently exceeding 0.993 and 0.995, respectively.

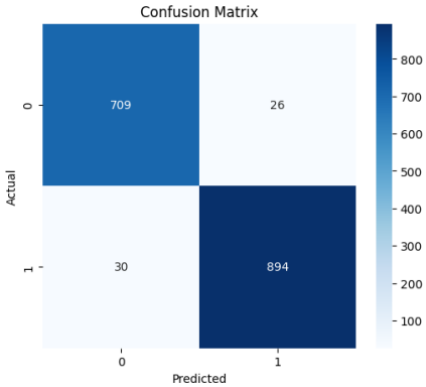

Figure 9. FT-Transformer with SMOTEN and PSO's Confusion Matrix

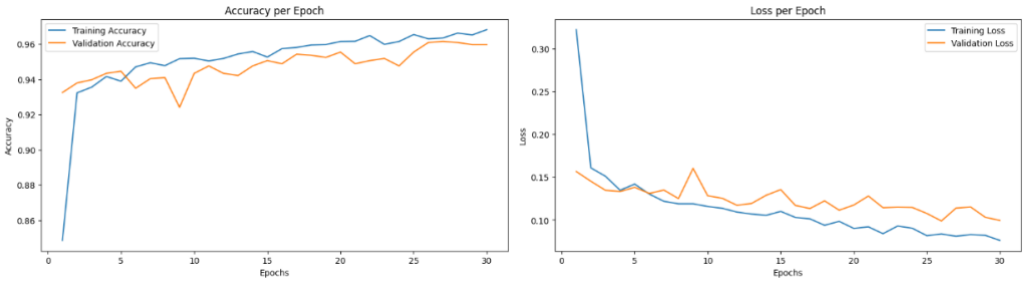

Figure 10. FT-Transformer with SMOTEN and PSO's Acc and Loss graph

The confusion matrix in Figure 9 illustrates an optimal balance, with both false positives and false negatives reduced compared with the previous two scenarios. The accuracy and loss curves in Figure 10 display the most stable training dynamics, showing consistent loss reduction without signs of overfitting. In Figure 11, the ROC curve lies closer to the ideal upper-left boundary than in the previous experiments, while the PRC curve highlights consistently high precision even at recall levels approaching 1.

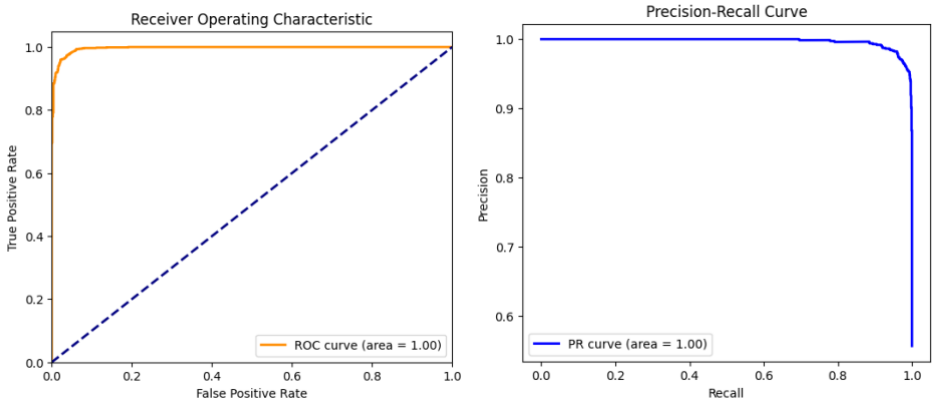

Figure 11. FT-Transformer with SMOTEN and PSO's ROC and PRC graphs

3.4 Comparative Discussion on Experiments

Table 4. 3 Cases Result Comparison

Case	Acc	F1-score	ROC-AUC	PR-AUC	Precision (Class 0 / 1)	Recall (Class 0 / 1)
Baseline FTT	95.36%	0.9579	0.99316	0.99456	0.94 / 0.97	0.96 / 0.95
FTT + SMOTEN	95.48%	0.9587	0.99361	0.99493	0.93 / 0.98	0.97 / 0.94
SMOTEN + FTT + PSO	96.62%	0.9696	0.99626	0.99694	0.96 / 0.97	0.96 / 0.97

As shown in Table 4, the three experiments conducted demonstrate a consistent trend of performance improvement. The baseline FT-Transformer (1st Experiment) already achieved a high ROC-AUC of 0.993 and PR-AUC of 0.994. However, the recall for the minority class lagged behind precision, as reflected in the confusion matrix in Figure 1, indicating a detection bias toward the majority class.

The introduction of SMOTEN (2nd Experiment) improved the balance of performance across classes. Although the gains in aggregate metrics such as accuracy and F1-score were relatively modest (approximately +0.1 pp and +0.0008, respectively), a notable improvement was observed in the recall for the minority class. The PRC curve in Figure 8 reveals a wider area in the high-recall region, indicating that the model maintained strong precision even as its sensitivity increased.

The third experiment, combining SMOTEN with PSO-based hyperparameter optimization (Experiment 3), delivered the most consistent and superior results. The F1-score reached 0.9696, while ROC-AUC rose to 0.9963 and PR-AUC approached 0.997. The confusion matrix (Figure 9) highlights an optimal balance between false positives and false negatives, whereas the ROC and PRC curves (Figs. 11 and 12) lie closer to the ideal boundary than in the previous two cases. These outcomes demonstrate that PSO tuning successfully aligned the representational capacity of the FT-Transformer with appropriate regularization, thereby achieving the best generalization performance. Overall, the experiments in Table 4 confirm that

integrating SMOTEN and PSO yields the most consistent and balanced performance, forming a solid basis for comparison with prior studies.

3.5 Comparative Analysis with Previous Studies

To contextualize the results obtained in this study, it is essential to compare the proposed framework with prior works reported in the literature. Table 5 summarizes the best-performing models from five related studies, including DNN [7], PCG with diffusion and multi-head attention[13], XGBoost optimized with Bat Algorithm [18], feature-selected Feedforward NN [33], and PhiDMA [34].

Table 5 shows a comparison in words between the method we suggest and other common ways to detect phishing that have been discussed in studies before. Because the studies being compared were tested on different sets of data with various numbers of examples, different ways of describing features, different ways the classes were distributed, different steps taken to prepare the data, and different methods used to conduct the experiments, the results they reported cannot be directly compared in terms of numbers. The table is given to show the differences in methods, the trends in performance that have been reported, and the advantages and disadvantages of each approach, so that the proposed framework can be properly placed in relation to what is currently known and used.

Table 5. Comparison with previous studies

Technique [IEEE]	Dataset	Acc (%)	Recall (%)	Precision (%)	F1	Strengths	Weaknesses
DNN [7]	Custom phishing dataset (Do et al., 2022)	96.38	97.13	96.52	96.83	Stable performance, effective in capturing complex patterns	Requires extensive tuning, prone to overfitting with limited data
PCG (Diffusion + MHA) [13]	Benchmark phishing dataset (imbalanced)	96.14	95.50	94.84	93.31	Effective in imbalanced datasets, combines generative and discriminative power	Computationally expensive, high training complexity
XGBoost + Bat Algorithm [18]	Multi-source phishing datasets	94.27	94.27	94.27	94.27	Adaptive hyperparameter tuning, strong for tabular data	Less effective than DL models in capturing deep non-linear relations
FNN (14 features) [33]	PhishTank (58,645 URLs → 14 selected features)	94.46	–	–	–	High accuracy with reduced features, computationally efficient	Highly dependent on feature engineering, limited adaptability to new data
PhiDMA [34]	PhishTank (667 phishing) + Phishload (995 legitimate)	92.72	90.54	91.23	~90.9	Multi-filter approach increases robustness (URL, whitelist, lexical)	Performance limited by small dataset, scalability issues
Proposed (SMOTEN + FT-Transformer + PSO)	PhishTank (automatic feature extraction, 30 features)	96.62	96.0–97.0	96.0–97.0	96.96	Balanced framework combining data resampling, Transformer architecture, and metaheuristic tuning; superior generalization	Higher computational cost (PSO), requires broader validation (cross-validation)

Even though the performance numbers in Table 5 can't be directly compared because the datasets, how the features are represented, the steps taken to prepare the data, and the setup of the experiments are different, the method we proposed still shows results that are pretty good compared to other recent ways of detecting phishing. The proposed framework achieved an accuracy of 96.62% and an F1-score of 96.96% when tested on the PhishTank dataset. It uses three main parts to improve performance: SMOTEN to handle imbalanced classes, FT-Transformer to learn from tabular data, and PSO to find the best settings for the model. Compared to earlier research, this new framework combines data resampling, Transformer-based learning, and metaheuristic optimization in a balanced way, which helps in effectively classifying phishing websites under the conditions tested in this study.

3.6 Limitations

Despite the substantial improvements achieved, several limitations should be acknowledged. First, the experiments were conducted with a single training seed; further validation using cross-validation or multiple seeds is required to confirm the robustness of the results. Second, although SMOTEN proved effective, the quality of synthetic samples warrants closer examination to ensure that they do not introduce

patterns that deviate from the original data distribution. Third, while PSO provided notable gains, it remains relatively computationally expensive; hybrid optimization approaches such as Bayesian optimization or multi-objective tuning could be explored in future work to improve efficiency.

4. CONCLUSION

This study was motivated by the challenges identified in the Introduction, namely the limited performance of phishing detection caused by imbalanced class distributions and the sensitivity of models to hyperparameter configurations. To address these issues, we proposed a hybrid framework integrating SMOTEN, the FT-Transformer, and Particle Swarm Optimization (PSO), and compared it against two reference scenarios: the baseline FT-Transformer and the SMOTEN-enhanced FT-Transformer. The experimental results consistently demonstrate that the proposed approach fulfilled the expectations set out in the Introduction. SMOTEN improved the model's sensitivity to the minority class by enhancing recall without substantially sacrificing precision, while PSO provided hyperparameter configurations that were better aligned with the characteristics of tabular phishing data, leading to more stable training dynamics and improved generalization. The full integration of the three components achieved the best performance, with an F1-score of 0.9696, a ROC-AUC of 0.9963, and a PR-AUC approaching 0.997, outperforming both comparison scenarios. These findings confirm that the synergistic integration of SMOTEN-based class balancing, the tabular FT-Transformer architecture, and PSO-based metaheuristic optimization effectively overcomes the weaknesses highlighted in the Introduction while enhancing the reliability of tabular phishing detection systems.

For future research, it is recommended to investigate more adaptive oversampling techniques, such as ADASYN or deep generative oversampling, as well as to explore the incorporation of Transformer-based ensemble methods to further improve the robustness of the model. In addition, calibrating the model's probability estimates could enhance interpretability and ensure dependable decision-making in real-world cybersecurity applications. In summary, this study not only contributes to advancing the theoretical foundation of deep learning–based phishing detection but also opens opportunities for developing more adaptive and practical frameworks to strengthen cybersecurity defenses in the future

REFERENCES

[1] Anti-Phishing Working Group, "apwg_trends_report_q1_2025," 2025. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2025.pdf

[2] IBM Institute for Business Value, "IBM: X-Force Threat Intelligence Index," 2022. doi: 10.12968/S1361-3723(22)70561-1.

[3] B. B. Gupta *et al.*, "A Hybrid CNN-Brown-Bear Optimization Framework for Enhanced Detection of URL Phishing Attacks," *Comput. Mater. Contin.*, vol. 81, no. 3, pp. 4853–4874, 2024, doi: 10.32604/cmc.2024.057138.

[4] M. Shariyab, "Phishing Website Detection," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 5, pp. 642–650, 2024, doi: 10.22214/ijraset.2024.61274.

[5] A. Alhuzali, A. Alloqmani, M. Aljabri, and F. Alharbi, "In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets," *Appl. Sci.*, vol. 15, no. 6, 2025, doi: 10.3390/app15063396.

[6] I. Abousaber, "A Novel Explainable Attention-Based Meta-Learning Framework for Imbalanced Brain Stroke Prediction," *Sensors*, vol. 25, no. 6, 2025, doi: 10.3390/s25061739.

[7] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[8] O. K. Sahingoz, E. Buber, and E. Kugu, "DEPHIDES: Deep Learning Based Phishing Detection System," *IEEE Access*, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[9] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data," 2023. doi: 10.1109/TNNLS.2021.3136503.

[10] S. K. Birthriya, P. Ahlawat, and A. K. Jain, "Intelligent phishing website detection: A CNN-SVM approach with nature-inspired hyperparameter tuning," *Cyber Secur. Appl.*, vol. 3, 2025, doi: 10.1016/j.csa.2025.100100.

[11] H. Fares *et al.*, "Machine Learning Approach for Email Phishing Detection," in *Procedia Computer Science*, 2024, pp. 746–751. doi: 10.1016/j.procs.2024.11.179.

[12] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and A. N. Saritha, "Phishing detection using random forest, SVM and neural network with backpropagation," in *Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics, ICSTCEE 2020*, 2020, pp. 391–394. doi: 10.1109/ICSTCEE49637.2020.9277256.

[13] X. Yang, J. Yuan, and N. Yang, "Phishing Website Detection on Imbalanced Datasets Based on Diffusion Model and Multi-Head Self-Attention," in *Proceedings of 2025 2nd International Conference on Communication Security and Information Processing, CSIP 2025*, 2025, pp. 1–6. doi: 10.1145/3744668.3744669.

[14] J. W. Lee, Y. E. Jeon, and J. I. Seo, "An integrated oversampling and noise reduction method for robust predictive analytics," *Decis. Anal. J.*, vol. 16, p. 100612, 2025, doi: 10.1016/j.dajour.2025.100612.

[15] S. Matharaarachchi, M. Domaratzki, and S. Muthukumarana, "Enhancing SMOTE for imbalanced data with abnormal minority instances," *Mach. Learn. with Appl.*, vol. 18, p. 100597, 2024, doi: 10.1016/j.mlwa.2024.100597.

[16] F. Danitasari, M. Ryan, D. Handoko, and I. Pramuwardani, "Improving Accuracy of Daily Weather Forecast Model at Soekarno-Hatta Airport Using BILSTM with SMOTE and ADASYN," *J. Penelit. Pendidik. IPA*, vol. 10, no. 1, pp. 179–193, 2024, doi: 10.29303/jppipa.v10i1.5906.

[17] D. J. Liu, G. G. Geng, and X. C. Zhang, "Multi-scale semantic deep fusion models for phishing website detection," *Expert Syst. Appl.*, vol. 209, 2022, doi: 10.1016/j.eswa.2022.118305.

[18] S. K. Birthriya, P. Ahlawat, and A. K. Jain, "Phishing Website Detection with XGBoost and Adaptive Hyperparameter Optimization using the Bat Algorithm," in *Procedia Computer Science*, 2025, pp. 1774–1782. doi: 10.1016/j.procs.2025.04.429.

[19] A. X. Wang and B. P. Nguyen, "TTVAE: Transformer-based generative modeling for tabular data generation," *Artif. Intell.*, vol. 340, 2025, doi: 10.1016/j.artint.2025.104292.

[20] S. Jaradat, M. Elhenawy, R. Nayak, A. Paz, H. I. Ashqar, and S. Glaser, "Multimodal Data Fusion for Tabular and Textual Data: Zero-Shot, Few-Shot, and Fine-Tuning of Generative Pre-Trained Transformer Models," *AI*, vol. 6, no. 4, 2025, doi: 10.3390/ai6040072.

[21] S. Asiri, Y. Xiao, and T. Li, "PhishTransformer: A Novel Approach to Detect Phishing Attacks Using URL Collection and Transformer," *Electron.*, vol. 13, no. 1, 2024, doi: 10.3390/electronics13010030.

[22] T. R. Alsenani, S. I. Ayon, S. M. Yousuf, F. B. K. Anik, and M. E. S. Chowdhury, "Intelligent feature selection model based on particle swarm optimization to detect phishing websites," *Multimed. Tools Appl.*, vol. 82, no. 29, pp. 44943–44975, 2023, doi: 10.1007/s11042-023-15399-6.

[23] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting Deep Learning Models for Tabular Data," 2021. [Online]. Available: https://github.com/yandex-research/rtdl

[24] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," 2012.

[25] S. A. Tyastama, T. G. Laksana, and A. B. Arifa, "Prediksi Penyakit Ginjal Kronis Menggunakan Hibrid Jaringan Saraf Tiruan Backpropagation dengan Particle Swarm Optimization," *J. Innov. Inf. Technol. Appl.*, vol. 3, no. 1, pp. 9–16, 2021, doi: 10.35970/jinita.v3i1.588.

[26] M. E. Hossen *et al.*, "Boosting Cervical Cancer Prediction Leveraging a Hybrid FT-Transformer Model," *IEEE Access*, vol. 13, pp. 26876–26896, 2025, doi: 10.1109/ACCESS.2025.3538566.

[27] R. Dewi, R. Sri hayati, A. Saleh, D. Y. Hakim Tanjung, and A. Jinan, "Enhancing Machine Learning Algorithm Performance for Pcos Diagnosis Using Smotenc on Imbalanced Data," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 11, no. 1, pp. 55–63, 2025, doi: 10.33480/jitk.v11i1.6676.

[28] A. X. Wang, V. T. Le, H. N. Trung, and B. P. Nguyen, "Addressing imbalance in health data: Synthetic minority oversampling using deep learning," *Comput. Biol. Med.*, vol. 188, 2025, doi: 10.1016/j.compbiomed.2025.109830.

[29] K. Omari and A. Oukhatar, "Advanced Phishing Website Detection with SMOTETomek-XGB: Addressing Class Imbalance for Optimal Results," in *Procedia Computer Science*, 2025, pp. 289–295. doi: 10.1016/j.procs.2024.12.031.

[30] H. Dai *et al.*, "FT-Transformer: Resilient and Reliable Transformer with End-to-End Fault Tolerant Attention," 2025. doi: 10.1145/3712285.3759853.

[31] I. A. Fares and M. Abd Elaziz, "FT-Transformer for Intrusion Detection in IoT Environment," *Bull. Fac. Sci. Zagazig Univ.*, vol. 2025, no. 1, pp. 114–123, 2025, doi: 10.21608/bfszu.2024.297682.1400.

[32] M. Imani, A. Beikmohammadi, and H. R. Arabnia, "Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels," *Technologies*, vol. 13, no. 3, 2025, doi: 10.3390/technologies13030088.

[33] G. S. Nayak, B. Muniyal, and M. C. Belavagi, "Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection and Deep Learning Models," *IEEE Access*, vol. 13, pp. 33308–33320, 2025, doi: 10.1109/ACCESS.2025.3543738.

[34] G. Sonowal and K. S. Kuppusamy, "PhiDMA – A phishing detection model with multi-filter approach," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 1, pp. 99–112, 2020, doi: 10.1016/j.jksuci.2017.07.005.