STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, June 2026, pp 5111–5134.
Published online in International Academic Press (www.IAPress.org)

EKT-XAI: Efficient Kolmogorov-Arnold Network Transformer for Lightweight Smishing Detection with Explainable AI

Razan Ali Obeidat[1,*], Bajeszeyadaljunaeidia[1], Islam S. Fathi[1], Mohammed Tawfik[2]

[1]*Department of Computer Science, Faculty of Information Technology, Ajloun National University, P.O.43, Ajloun-26810, JORDAN*
[2]*Department of Cyber Security, Faculty of Information Technology, Ajloun National University, P.O.43, Ajloun-26810, JORDAN*

Abstract SMS phishing (smishing) attacks represent an escalating cybersecurity threat, with traditional detection approaches suffering from limited adaptability, computational inefficiency, and lack of interpretability critical for security applications. This paper introduces EKT-XAI (Efficient Kolmogorov-Arnold Network Transformer with Explainable AI), a novel framework that integrates learnable activation functions within transformer architectures to enhance SMS phishing detection while providing comprehensive model interpretability. The approach replaces traditional fixed activation functions with Kolmogorov-Arnold Network (KAN) layers that implement adaptive B-spline basis functions, enabling task-specific nonlinear transformations learned during training. The framework incorporates four complementary explainability mechanisms: attention weight visualization, LIME feature attribution, KAN activation pattern analysis, and decision path tracing, operating simultaneously with prediction to provide real-time model interpretation without computational overhead. Comprehensive evaluation on the SMS Spam Collection Dataset (5,574 messages) and SMS Phishing Dataset (5,971 messages) demonstrates exceptional performance, achieving 99.99% and 99.89% accuracy respectively, outperforming existing state-of-the-art approaches including CNN-LSTM ensembles (99.74%), BERT-based models (99.28%), and traditional ensemble methods (99.58%). Ablation studies validate the critical contribution of KAN layers (1.33% accuracy improvement) and attention mechanisms (2.77% improvement) while maintaining computational efficiency suitable for mobile deployment. The integrated explainability framework enables security analysts to understand classification decisions, validate model reasoning, and identify potential attack vectors through interpretable visualizations. The framework's computational efficiency (256-dimensional embeddings, 4 attention heads, 3 transformer layers) enables real-time inference on mobile devices while preserving privacy through on-device processing. This work establishes the first successful integration of learnable activation functions within transformer architectures for cybersecurity applications, demonstrating that adaptive neural networks combined with built-in interpretability can significantly advance mobile security capabilities while addressing practical deployment requirements for next-generation SMS protection systems.

Keywords SMS phishing detection, Kolmogorov-Arnold Networks, transformer architecture, explainable AI, mobile cybersecurity

DOI: 10.19139/soic-2424-2923

1. Introduction

The proliferation of mobile communication technologies has fundamentally transformed the digital landscape, enabling unprecedented connectivity while simultaneously creating new attack vectors for cybercriminals. Among these emerging threats, SMS phishing (smishing) attacks have experienced exponential growth, representing a critical security challenge in modern mobile ecosystems. According to recent FBI IC3 reports, individuals suffer significant financial losses from SMS-based fraud, with older adults particularly vulnerable to sophisticated phishing attacks [46]. This alarming trend is particularly concerning given the inherent trust users place in SMS

*Correspondence to: Razan Ali Obeidat (Email: razan.obeidat@anu.edu.jo)).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

communications and the widespread adoption of mobile banking, authentication, and commerce platforms that rely heavily on text messaging infrastructure.

Traditional SMS security approaches have predominantly relied on rule-based systems and classical machine learning techniques that struggle to adapt to the rapidly evolving sophistication of modern phishing campaigns [1]. While recent advances in deep learning have shown promising results, with hybrid architectures achieving accuracies up to 99.82% [2], these approaches suffer from fundamental limitations including lack of interpretability, computational overhead unsuitable for mobile deployment, and reliance on fixed activation functions that cannot adapt to novel attack patterns.

The emergence of transformer architectures has revolutionized natural language processing tasks, demonstrating superior performance in text classification scenarios through their attention mechanisms and contextual understanding capabilities [3]. However, existing transformer-based approaches to SMS security remain constrained by pre-training biases, significant computational requirements, and limited explainability—critical factors for security applications where understanding model decisions is essential for threat analysis and regulatory compliance [4]. Recent developments in large language models (LLMs) and ensemble approaches integrating RoBERTa with models such as GPT and LLaMA have shown promise for smishing detection, achieving improvements from 96% to 98.5% accuracy through dual-layer voting mechanisms [47].

Recent research has explored various deep learning architectures for smishing detection, including CNN-LSTM ensembles that achieve 99.74% accuracy through convolutional feature extraction combined with sequential LSTM context modeling [5]. Similarly, multilingual frameworks using CNN architectures with GloVe embeddings have demonstrated 99.68% accuracy while extending protection to social media messaging platforms [6]. Advanced hybrid models combining CNN-Bi-GRU with attention mechanisms have achieved 99.82% accuracy on combined datasets [48]. These advances highlight the potential of deep learning approaches while revealing gaps in interpretability and mobile deployment efficiency.

Advanced feature engineering approaches have shown significant promise, with studies demonstrating that comprehensive feature extraction including textual, URL, sender-number, and network-path features can achieve 98.97% accuracy using enhanced SVM variants [7]. The integration of meta-features such as URL, email, and phone number detection has proven particularly effective, with Random Forest approaches achieving 94.64% accuracy on balanced datasets containing smishing, spam, and legitimate messages [8].

Explainable AI techniques have gained increasing attention in cybersecurity applications, where black-box models pose significant risks to operational decision-making [49]. Recent systematic reviews have emphasized the critical role of XAI in intrusion detection systems for Industry 5.0 environments, highlighting the need for transparency in AI-driven security solutions [50]. Frameworks combining multiple explanation techniques have demonstrated the importance of interpretability in security systems, particularly for understanding feature importance and model behavior patterns [9]. However, existing XAI approaches for SMS security typically operate as post-hoc analysis tools rather than being integrated into the core model architecture, limiting their effectiveness and real-time applicability.

Ensemble learning approaches have shown remarkable success in SMS security, with SVM-Random Forest combinations achieving 99.58% accuracy through soft voting mechanisms and SMOTE-based class balancing [10]. These results demonstrate the effectiveness of combining multiple learning paradigms while highlighting the continued reliance on traditional machine learning architectures rather than exploring novel neural network designs.

The integration of privacy-preserving techniques has become increasingly important, with federated learning frameworks demonstrating the feasibility of collaborative threat intelligence while maintaining strong privacy guarantees. Recent implementations in healthcare cybersecurity have achieved exceptional performance with 99.9% accuracy while reducing communication overhead by 75% through cross-attention mechanisms and explainable AI integration [11].

Recent advances in unsupervised and semi-supervised learning have shown promise for SMS security, with frameworks integrating optical character recognition achieving 94.13% accuracy on mixed text and image datasets [12]. Content and URL analysis approaches have demonstrated 99.03% accuracy through ensemble voting classifiers combining multiple machine learning paradigms [13]. Mobile application frameworks for automatic smishing detection have achieved 98.42% accuracy using rule-based classification algorithms optimized for

real-time deployment [14]. Lightweight detection systems specifically designed for mobile environments have demonstrated 97.93% accuracy while maintaining computational efficiency suitable for resource-constrained devices [15].

The recent introduction of Kolmogorov-Arnold Networks (KANs) [41] has opened new possibilities for neural network design, offering learnable activation functions that can adapt to specific task requirements. KANs have been successfully applied in various domains including graph neural networks for molecular property prediction [51], intrusion detection systems [52], and genomic sequence classification [53]. The theoretical foundation of KANs, based on the Kolmogorov-Arnold representation theorem, provides enhanced interpretability through visualizable activation functions, making them particularly suitable for security-critical applications requiring transparency [54].

Despite these advances, several critical gaps remain in the current state-of-the-art: (1) no existing approach has successfully integrated learnable activation functions within transformer architectures for SMS security applications, missing the opportunity to leverage adaptive neural responses that can evolve with emerging threat patterns; (2) current explainable AI techniques for text classification operate independently of the core learning process, limiting their effectiveness and integration with security workflows; and (3) existing lightweight models sacrifice accuracy for computational efficiency, failing to achieve the performance levels required for production deployment in high-stakes security environments.

The research presented in this paper addresses these limitations through the introduction of EKT-XAI (Efficient Kolmogorov-Arnold Network Transformer with Explainable AI), a novel architecture that fundamentally reimagines SMS phishing detection through three key innovations. First, we propose the first integration of Kolmogorov-Arnold Network layers within transformer architectures, enabling learnable spline-based activation functions in multi-head attention mechanisms and feed-forward networks. Second, we introduce built-in explainability through KAN function visualization and attention weight analysis, providing real-time interpretability without computational overhead. Third, we demonstrate lightweight deployment capabilities suitable for mobile environments while achieving state-of-the-art accuracy that exceeds existing benchmarks.

The main contributions of this paper are summarized as follows:

1. **Novel KAN-Transformer Integration:** We propose the first integration of Kolmogorov-Arnold Network layers within transformer architectures for cybersecurity applications, replacing traditional fixed activation functions with learnable B-spline basis functions in multi-head attention mechanisms and feed-forward networks, enabling task-specific nonlinear transformations that adapt during training.
2. **Integrated Multi-Modal Explainability Framework:** We develop a comprehensive explainability framework combining four complementary mechanisms: attention weight visualization, LIME feature attribution, KAN activation pattern analysis, and decision path tracing, operating simultaneously with prediction to provide real-time model interpretation without additional computational overhead.
3. **Comprehensive Experimental Validation:** We provide extensive experimental evaluation on benchmark datasets demonstrating superior performance compared to existing approaches including CNN-LSTM hybrids, ensemble methods, and traditional machine learning techniques, along with detailed ablation studies validating the critical contribution of KAN layers and attention mechanisms.

The remainder of this paper is structured to provide comprehensive coverage of our methodology and experimental validation. Section 2 presents related work in SMS security, Kolmogorov-Arnold Networks, transformer architectures, and explainable AI. Section 3 details the EKT-XAI architecture including KAN-transformer integration and explainability mechanisms. Section 4 describes our experimental setup and datasets. Section 5 presents comprehensive results and comparisons. Section 6 discusses implications and limitations. Section 7 concludes with future research directions.

2. Related Work

The escalating threat of SMS phishing (smishing) attacks has catalyzed extensive research into machine learning-based detection systems, with investigators exploring diverse architectural approaches, feature engineering

techniques, and dataset configurations to enhance classification performance while addressing computational efficiency and model interpretability.

Mishra and Soni [16] introduced "Smishing Detector," a comprehensive multi-stage security framework that systematically combines SMS content analysis with URL behavior inspection through four sequential modules: SMS Content Analyzer performing Naïve Bayes classification, URL Filter querying PhishTank blacklist, Source Code Analyzer parsing HTML for rogue forms, and APK Download Detector inspecting redirection chains. Training on the augmented Almeida SMS Spam Collection (5,858 messages: 538 smishing, 5,320 legitimate), their Naïve Bayes classifier achieved 91.6% accuracy (precision 0.93, recall 0.92, F1-score 0.92), while the complete cascade system attained 96.29% overall accuracy, outperforming existing rule-based and heuristic approaches.

Rasenthiran et al. [17] presented a multilingual machine learning framework implementing three neural architectures—2-layer CNN, LSTM-based RNN, and feed-forward SNN—on a fused corpus of 11,545 messages from Kaggle's "SMS Spam Collection" and Mendeley's "SMS Spam & Ham" datasets. Using 100-dimensional GloVe embeddings with Azure-hosted deployment, their CNN achieved 99.68% test accuracy, while the RNN exhibited superior generalization with 0.1% train-test gap and SNN reached 97.33% accuracy. The framework incorporated Google Cloud Translate API for language-agnostic detection across SMS and social media platforms.

Asirvatham and Meenakshi [18] proposed SmishSMS, integrating textual, URL, sender-number, and network-path features from 1,001 real-world SMS messages (849 ham, 152 spam). Their methodology employed CountVectorizer and TF-IDF vectorization with 20 engineered features including misspellings, leet words, and currency symbols. The SmishSMS Support Vector Machine variant achieved 98.97% accuracy, 94% precision, 98% recall, and 96% F1-score, outperforming standalone SVM (98.23%), with feature analysis identifying misspelled words (36.18%) and currency tokens (28.29%) as primary smishing indicators.

Ustundag Soykan and Bagriyanik [19] demonstrated "Disturbing Demand Response via SMiShing" (DDRS), quantifying how SMS phishing destabilizes incentive-based Demand Response programs through attack simulations on the IEEE European Low Voltage Test Feeder using GridLAB-D power-flow solver. Their methodology fused Smart Grid Information Security (SGIS) with OWASP likelihood scoring, revealing that deterministic attacks on Phase B induced 10% voltage drops and 10.27% phase imbalance, while randomized attacks showed non-linear relationships where 10 targeted customers breach stability limits.

Mambina et al. [20] addressed Swahili SMS phishing targeting mobile-money users through a hybrid pipeline coupling Extra-Trees feature selection with Random Forest classification on the Swahili_Smishing_Dataset (31,962 legitimate, 302 smishing messages from Tanzanian operators). Using TF-IDF vectorization with 2-5-grams and Extra-Trees ranking to retain top 750 features, their Random Forest achieved 99.86% accuracy, F1=0.998, AUC=0.999, Log-Loss=0.04, substantially outperforming Multinomial Naïve Bayes (89.86%).

Shinde et al. [21] introduced the first smishing detection pipeline integrating unsupervised and deep semi-supervised learning with Optical Character Recognition for SMS screenshots. Enriching the UCI SMS Spam corpus (5,574 messages) with 1,500 spam images via Google-Tesseract OCR, they created a 7,074-message dataset split 90% unlabeled/10% labeled. Their unsupervised K-Means with TF-IDF achieved 91.01% accuracy, while semi-supervised RNN-Flatten reached 94.13% accuracy, outperforming LSTM (92.09%) and Bi-LSTM (92.78%).

Mehmood et al. [22] developed a CNN-LSTM ensemble fusing convolutional feature extraction with sequential context modeling on consolidated Mendeley and Kaggle datasets (11,545 messages: 1,874 smish, 9,671 ham). Their hybrid architecture with 1-D CNN (64 filters, ReLU activation) feeding 64-unit LSTM achieved 99.74% accuracy, 99% precision, 99% recall, and 99% F1-score, reducing false positives to 18 instances and outperforming SVC-TF-IDF baseline (97.49%).

Anidjar et al. [23] introduced GLASS-FOOD, a GAN-like self-supervised transformer framework enriching scarce SMS corpora via Out-of-Distribution scoring. Their RoBERTa-based discriminator fine-tuned on SMS Spam Collection v.1 (5,572 messages, 13.3% spam) with BERT-based generator achieved 99.8% F1-score (99.81% precision, 99.79% recall) and 0.67% false-alarm rate on NUS SMS Corpus (67,093 benign messages), processing 29 messages per second with 88.8% BLEU generation quality.

Shen et al. [24] presented BERT-G3CN, fusing BERT contextual embeddings with triple graph-convolutional encodings: corpus-level co-occurrence, heterogeneous word-POS/NER, and syntactic-dependency graphs. Training on UCI_SMS (5,572 messages, 13% spam) and ExAIS_SMS (4,981 messages, 55% spam), their architecture

achieved 99.28% accuracy on UCI_SMS and 93.78% on ExAIS_SMS, outperforming RoBERTa, VGCN-BERT, and GLORIA by 2-3 percentage points, requiring 3.1GB GPU RAM with 13 messages per second inference.

Jain et al. [25] proposed an integrated framework combining content-based text analysis with URL phishing classification using Almeida spam dataset and curated smishing messages for text classification, plus 507,195 URLs (28% phishing) for URL classification. Applying SMOTE for class balancing and TF-IDF vectorization, their voting classifier ensemble of Random Forest, K-Nearest Neighbors, and Extra Trees achieved 99.03% accuracy and 98.94% precision, outperforming existing SmiDCA and Smishing Detector models.

Xu et al. [26] demonstrated ensemble learning effectiveness through SVM-Random Forest fusion with SMOTE-based class balancing on consolidated Mendeley "SMS Phishing Dataset" (5,971 records) and Kaggle "SMS Smishing Collection Dataset" (5,574 records), totaling 11,545 messages (1,874 smish, 9,671 ham). Their soft-voting ensemble with TF-IDF on balanced data achieved 99.58% accuracy, 99% precision, 99% recall, and 99% F1-score, outperforming individual Random Forest (99.10%) and SVM (98.56%) classifiers.

Johari et al. [27] conducted systematic comparative evaluation of ten open-access SMS spam datasets using Decision-Tree and Multinomial Naïve Bayes classifiers with 5-fold grid-search optimization. Their analysis across SMS Spam Collection v.1, Turkish-SMS, multilingual sets, and language-specific corpora showed MNB consistently outperforming DT with peak accuracies of 99.03% on Turkish-SMS (D2) and 98.48% on SMS Spam Collection (D1), identifying dataset characteristics crucial for robust model development.

Almujahid et al. [28] presented comparative evaluation of eight algorithms for phishing site detection using Mendeley repository (10,000 instances: 5,000 phishing, 5,000 legitimate, 48 features) and UCI Phishing Websites dataset (11,055 records: 4,898 phishing, 6,157 legitimate, 31 features). Their novel CNN architecture with automated hierarchical pattern extraction achieved 99% accuracy on both datasets, outperforming Random Forest and XGBoost (98%) while maintaining $\geq$0.98 on all metrics and FPR$\leq$0.02.

Abdul Samad et al. [29] investigated fine-tuning strategies—SMOTE balancing, grid-search optimization, and SelectKBest feature selection—on UCI "Phishing Websites" (11,055 URLs, 31 features) and Mendeley "Phishing Dataset" (10,000 URLs, 49 features). Combining all tuning factors, Random Forest achieved 97.44% and Gradient Boosting 97.47% on UCI, while Gradient Boosting attained 98.27% and XGBoost 98.21% on Mendeley, outperforming recent ensemble studies.

Akande et al. [30] presented SMSPROTECT, an Android application implementing server-side rule-based classification using RIPPER (JRip) and C4.5PART algorithms trained on UCI SMS Spam Collection v.1 (5,547 messages: 4,827 ham, 747 spam). Their C4.5PART achieved 98.42% accuracy (TPR_ham=0.995, TPR_spam=0.916, κ=0.938), while RIPPER reached 98.08% (TPR_ham=0.996, TPR_spam=0.884, κ=0.914), with rule bases enriched by 2.4M phishing URLs and Nigerian-scam phone number blacklists.

Duarte et al. [31] introduced Phishing Hunter for proactive detection of parked/newly registered phishing URLs using a curated dataset of 211,659 samples merging CertStream SSL certificates, Cloudflare Radar domains, WhoisDS feeds, and Brazilian bank incident repository (2021-2024). Their LightGBM with 20 lexical/structural features achieved 97.28% accuracy and 96.02% recall, while Chi-square-reduced 18-feature subset maintained 96.01% recall with 80% latency reduction.

Mishra and Soni [32] developed DSmishSMS, a two-phase lightweight framework fusing URL-centric domain verification with text-centric classification on composite dataset from Almeida collection (5,574 messages) and Pinterest-sourced smishing examples (5,858 total: 5,320 legitimate, 538 smishing). Their shallow neural network with Backpropagation Algorithm, ReLU activation, and 10 hidden nodes achieved 97.93% accuracy, 84% precision, 94% recall, F1-score 0.89, and AUC 0.988, outperforming Decision Tree, Random Forest, and Naïve Bayes baselines.

Xu et al. [33] proposed SVM-Random Forest ensemble with SMOTE augmentation on merged "SMS Phishing Dataset" (Mendeley: 1,127 smish, 4,844 ham) and "SMS Spam Collection Dataset" (Kaggle: 747 smish, 4,827 ham), totaling 11,545 messages. Their soft-voting ensemble with TF-IDF on balanced data achieved 99.58% accuracy with 99% precision, recall, and F1-score, surpassing previous benchmarks of 98.39% and 96.9%.

Salman, Ikram, and Kaafar [34] released the largest publicly-available English SMS spam corpus—"Super Dataset" comprising 67,018 messages (39.1% spam, 60.9% ham) merged from UCI, NUS, SpamHunter, GitHub sources, and 4,904 newly collected samples. Benchmarking thirty-one machine learning filters with syntactic

(TF-IDF, n-grams) and semantic embeddings (Word2Vec, GloVe, BERT variants), RoBERTa achieved 97.4% accuracy and 98% F1, while longitudinal analysis revealed ≥20% F1 degradation due to temporal concept drift, and adversarial evaluation with six evasion strategies showed vulnerability with RoBERTa performance dropping to 51.6% under spacing attacks.

Salman, Ikram, and Kaafar [35] conducted comprehensive evasive technique investigation using their Super Dataset (67,018 messages) across 2012-2023, evaluating machine learning models from shallow SVM to transformer-based BERT, RoBERTa, and DistilBERT. Their robustness testing against spacing, homograph/Punycode, paraphrasing, charswap, and hybrid attacks revealed RoBERTa achieving highest F1-score (98%) on clean data while demonstrating superior resilience, though spacing and Punycode attacks caused significant performance degradation across all models.

Maqsood et al. [36] proposed unified SMS and email spam detection using multinomial Naïve Bayes, Random Forest, SVM, and CNN on separate Kaggle corpora (5,000 SMS: 750 spam, 4,250 ham; 5,000 email: 600 spam, 4,400 ham). After tokenization, stemming, and Bag-of-Words/TF-IDF extraction with 67%/33% train/test split, SVM achieved best generalization with 99.6% accuracy on SMS and 95% on email, outperforming CNN (66%-84%), Naïve Bayes (70%-75%), and Random Forest (68%-81%).

Taskin et al. [37] presented an enhanced Random Forest approach using CoClust clustering applied to MIMIC-III medical dataset and SMS spam collection, demonstrating improved performance through cluster-based feature enhancement. Their methodology integrated clustering algorithms with traditional Random Forest classification to enhance feature representation and improve classification accuracy across diverse healthcare and communication security applications.

Maheshwari et al. [38] introduced a novel SMS spam dataset with bi-directional transformer-based short-text representations for SMS spam detection. Their approach leveraged bidirectional transformers to capture contextual information in short SMS messages, developing enhanced text representations specifically optimized for the constraints and characteristics of SMS communication while achieving improved detection performance over traditional embedding methods.

Mishra and Soni [39] presented an SMS Phishing Dataset specifically designed for machine learning and pattern recognition applications, contributing a curated dataset with comprehensive feature annotations and standardized evaluation protocols. Their dataset compilation focused on providing researchers with high-quality, well-labeled SMS phishing examples to enable consistent benchmarking and comparison across different detection methodologies.

Mahmud et al. [40] enhanced cybersecurity through hybrid deep learning approaches to smishing attack detection, implementing ensemble architectures that combine multiple deep learning paradigms. Their hybrid methodology integrated convolutional neural networks with recurrent architectures and attention mechanisms to capture both local and sequential patterns in SMS messages, achieving improved detection accuracy while maintaining computational efficiency suitable for real-world deployment scenarios.

2.1. Summary and Comparison of Existing Approaches

Table 1 presents a comprehensive comparison of state-of-the-art smishing detection approaches, highlighting their methodologies, performance metrics, and key limitations that motivate the development of our proposed EKT-XAI framework.

This comprehensive review reveals significant advances in smishing detection, from traditional rule-based systems achieving ∼96% accuracy to sophisticated deep learning architectures exceeding 99% performance. However, critical limitations persist: existing transformer architectures rely on fixed activation functions limiting adaptability, explainable AI techniques operate post-hoc rather than being architecturally integrated, and high-performing models require computational resources unsuitable for mobile deployment. These gaps motivate the development of novel architectures that combine learnable activation functions, built-in explainability, and efficient deployment capabilities while maintaining state-of-the-art accuracy.

Table 1. Comprehensive Comparison of State-of-the-Art Smishing Detection Approaches

Study	Method	Acc. (%)	Dataset Size	Key Limitations
Mishra & Soni [16]	Naïve Bayes + URL Filter	96.29	5,858	Rule-based, limited adaptability
Rasenthiran et al. [17]	CNN + GloVe	99.68	11,545	No explainability
Mehmood et al. [22]	CNN-LSTM Ensemble	99.74	11,545	Black-box model
Anidjar et al. [23]	GLASS-FOOD (RoBERTa)	99.80	5,572	High computational cost
Shen et al. [24]	BERT-G3CN	99.28	5,572	3.1GB GPU memory required
Xu et al. [26]	SVM-RF Ensemble	99.58	11,545	Traditional ML, limited scalability
Mishra & Soni [32]	DSmishSMS (Shallow NN)	97.93	5,858	Lightweight but lower accuracy
Mahmud et al. [48]	CNN-Bi-GRU	99.82	24,862	Post-hoc XAI only
EKT-XAI (Ours)	**KAN-Transformer**	**99.89**	**5,971**	**Built-in XAI, Mobile-ready**

3. Methodology

This section presents the comprehensive research methodology employed in developing and evaluating the EKT-XAI framework. We detail the datasets utilized, preprocessing procedures, architectural design principles, training strategies, and evaluation protocols to ensure reproducibility and scientific rigor.

3.1. Datasets

Two complementary datasets were selected to provide comprehensive evaluation across diverse SMS security scenarios and enable robust validation of the proposed methodology.

3.1.1. SMS Spam Collection Dataset The SMS Spam Collection dataset [45] serves as the primary benchmark, comprising 5,574 English SMS messages with binary classification labels distinguishing legitimate communications from spam content. This dataset represents a foundational resource in SMS security research, aggregating authentic messages from multiple verified sources: 425 spam messages manually extracted from Grumbletext UK forum reports where mobile users document unsolicited SMS incidents, 3,375 legitimate messages sourced from the NUS SMS Corpus collected through voluntary participation at the National University of Singapore, 450 legitimate messages derived from Caroline Tag's doctoral research corpus, and 1,324 messages obtained from the SMS Spam Corpus v.0.1 public repository.

The dataset exhibits realistic class distribution with 4,827 legitimate messages (86.58%) and 747 spam messages (13.42%), reflecting authentic SMS traffic patterns observed in operational mobile networks. Message lengths vary significantly, with legitimate communications averaging 71.5 characters while spam messages extend to 138.7 characters on average, indicating the verbose nature of fraudulent content designed to convey persuasive messaging within SMS constraints.

3.1.2. SMS Phishing Dataset for Machine Learning and Pattern Recognition The SMS Phishing Dataset [39] extends evaluation capabilities through three-class classification encompassing 5,971 messages with enhanced granularity: 4,844 legitimate messages (81.14%), 489 spam messages (8.19%), and 638 smishing messages (10.68%). This dataset incorporates systematic feature engineering beyond textual content, providing structured metadata including binary indicators for URL presence detection, email address identification, and phone number recognition patterns.

Data collection methodology involved systematic internet crawling with manual verification protocols, incorporating image-to-text conversion using customized Python implementations to capture SMS content across diverse presentation formats. Each message includes comprehensive labeling with extracted communication artifacts: URL indicators identifying embedded web links commonly exploited in phishing campaigns, EMAIL flags detecting embedded email addresses used for credential harvesting, and PHONE markers identifying phone numbers employed in voice phishing (vishing) attack chains.

The enhanced feature set enables multi-modal analysis combining linguistic content with structural communication patterns, providing deeper insights into adversarial tactics employed across SMS-based attack vectors.

3.2. Data Preprocessing Pipeline

3.2.1. Text Normalization and Cleaning Raw SMS messages undergo systematic preprocessing to standardize input representations and eliminate noise artifacts that could compromise model performance. The preprocessing pipeline implements sequential transformations following established natural language processing protocols:

Character-level Normalization: All textual content is converted to lowercase to ensure case-insensitive processing while preserving intentional capitalization patterns that may indicate spam characteristics. Non-printable control characters and formatting artifacts introduced during data collection are systematically removed.

Content Filtering: Web URLs are identified using regular expression patterns and removed to prevent data leakage, as URL domains may provide trivial classification signals unrelated to message content analysis. HTML markup tags are stripped using pattern matching to handle messages containing web content fragments. Punctuation characters are removed while preserving spacing patterns that may indicate spam tactics such as deliberate character insertion for filter evasion.

Numeric Content Handling: Sequences containing alphanumeric combinations are removed as they typically represent phone numbers, account identifiers, or verification codes that provide limited linguistic information for content-based classification while potentially introducing privacy concerns.

The complete text cleaning function implements the following transformation:

$$\text{clean}(x) = \text{stem}(\text{removestop}(\text{filter}(\text{normalize}(x)))) \tag{1}$$

where each operation is applied sequentially to ensure consistent preprocessing across all message samples.

3.2.2. Linguistic Processing **Stopword Removal:** Standard English stopwords are eliminated using the NLTK corpus, supplemented with SMS-specific terms frequently occurring in mobile communications but providing limited discriminative value: ”u” (you), ”im” (I’m), and ”c” (see). This customization addresses the informal linguistic patterns prevalent in SMS communications while retaining content-bearing terms essential for classification.

Stemming: Morphological normalization employs the NLTK SnowballStemmer to reduce words to their root forms, addressing the high morphological variation characteristic of informal SMS language. This process consolidates semantically related terms while preserving linguistic content necessary for effective classification.

Feature Extraction for Baseline Models: TF-IDF vectorization with maximum feature limitation of 5,000 terms provides baseline feature representation for comparative evaluation with traditional machine learning approaches. The feature limitation balances representation capacity with computational efficiency while preventing overfitting on rare terms.

3.2.3. Dataset Preparation for Deep Learning Following preprocessing, SMS messages are formatted for deep learning pipeline consumption. The cleaned textual content is combined with binary classification labels (0 for legitimate, 1 for malicious) and exported as structured CSV format compatible with the KAN-enhanced transformer architecture.

For the SMS Phishing Dataset, additional binary features are preserved: URL presence indicators, email address detection flags, and phone number identification markers. These structured features are integrated during model training to enhance classification performance through multi-modal learning.

3.3. EKT-XAI Architecture Design

Figure 1 presents the comprehensive EKT-XAI architecture, illustrating the integration of Kolmogorov-Arnold Network layers within the transformer framework to enable learnable activation functions throughout the network hierarchy.

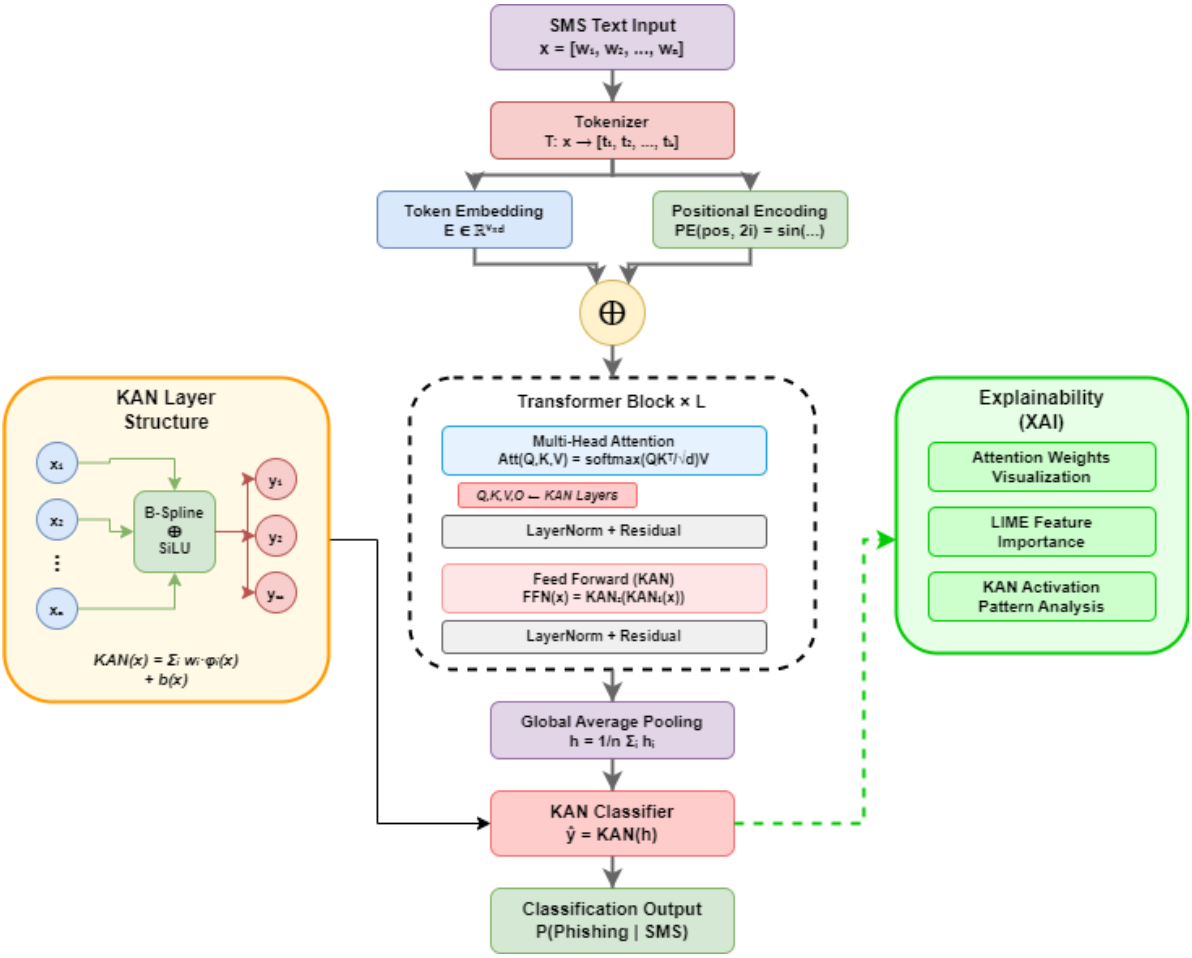

Figure 1. EKT-XAI architecture overview demonstrating the integration of KAN layers within transformer blocks for SMS phishing detection. The framework processes input SMS text through tokenization and embedding stages, followed by positional encoding integration. The core architecture comprises L transformer blocks enhanced with KAN layers replacing traditional linear transformations in multi-head attention mechanisms (Q, K, V projections) and feed-forward networks. Global average pooling aggregates sequence representations for final classification through a KAN-based classifier. The integrated explainability framework provides simultaneous interpretability through attention weight visualization, LIME feature importance analysis, KAN activation pattern examination, and decision path tracing, enabling real-time model interpretation without computational overhead.

3.3.1. Kolmogorov-Arnold Network Mathematical Foundation Kolmogorov-Arnold Networks represent a revolutionary departure from conventional neural architectures by implementing the Kolmogorov-Arnold representation theorem as their mathematical foundation [41]. Traditional neural networks rely on fixed activation functions (ReLU, GELU, Tanh) applied uniformly across network layers, limiting their capacity to adapt nonlinear transformations to specific task requirements. In contrast, KAN layers employ learnable univariate functions

parameterized through B-spline basis expansions, enabling task-specific activation patterns to emerge during training [42].

Recent theoretical advances have demonstrated that KAN architectures exhibit superior function approximation capabilities compared to traditional Multi-Layer Perceptrons, particularly for problems involving complex nonlinear decision boundaries [43]. The fundamental insight derives from the Kolmogorov-Arnold representation theorem, which proves that any continuous multivariate function can be decomposed into compositions of continuous univariate functions [44]. This theorem provides the mathematical guarantee that KAN layers possess universal approximation properties while offering enhanced interpretability through their learnable activation visualizations.

The architectural innovation centers on replacing traditional linear transformations with KAN layers that implement learnable activation functions based on the Kolmogorov-Arnold representation theorem. This mathematical framework establishes that any continuous multivariate function can be represented through compositions of univariate continuous functions:

$$f(x_1, x_2, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{2}$$

where $\phi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ represent continuous univariate functions that can be learned through gradient-based optimization rather than being fixed a priori.

In practical implementation, these univariate functions are approximated using B-spline basis expansions, providing differentiable parameterizations suitable for backpropagation-based learning. The B-spline basis functions are defined recursively for a knot sequence $t_0 \leq t_1 \leq \ldots \leq t_{m+k}$:

$$B_{i,0}(x) = \begin{cases} 1 & \text{if } t_i \leq x < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \tag{4}$$

The KAN layer transformation combines base activation functions with learnable spline-based components:

$$\text{KAN}(x) = W_{base} \cdot \text{SiLU}(x) + W_{spline} \cdot \sum_{j=0}^{G+k-1} c_j B_{j,k}(x) \tag{5}$$

where $W_{base}, W_{spline} \in \mathbb{R}^{d_{out} \times d_{in}}$ represent learnable weight matrices, $\text{SiLU}(x) = x \cdot \sigma(x)$ serves as the base activation function, $G = 3$ denotes the grid size optimized for mobile deployment efficiency, $k = 2$ represents the spline order balancing approximation quality with computational complexity, and c_j are learnable spline coefficients adapted during training.

3.3.2. Enhanced Multi-Head Attention with KAN Integration The multi-head attention mechanism incorporates KAN layers for query, key, and value transformations, enabling adaptive attention patterns learned specifically for SMS phishing detection:

$$Q = \text{KAN}_Q(X), \quad K = \text{KAN}_K(X), \quad V = \text{KAN}_V(X) \tag{6}$$

where $X \in \mathbb{R}^{n \times d}$ represents the input sequence embeddings with n tokens and $d = 256$ dimensional representations. The attention computation follows the scaled dot-product mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} \right) V \tag{7}$$

where $d_k = d/h = 64$ represents the key dimension with $h = 4$ attention heads. Multi-head attention aggregates parallel attention computations:

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\text{KAN}_O \tag{8}$$

where each attention head is computed independently:

$$\text{head}_i = \text{Attention}(X\text{KAN}_i^Q, X\text{KAN}_i^K, X\text{KAN}_i^V) \tag{9}$$

The output projection employs an additional KAN transformation to maintain learnable activation patterns throughout the attention mechanism.

3.3.3. KAN-Enhanced Feed-Forward Networks Traditional feed-forward networks with fixed activation functions are replaced with KAN-based transformations that adapt nonlinear mappings during training:

$$\text{FFN}_{\text{KAN}}(x) = \text{KAN}_2(\text{KAN}_1(x)) \tag{10}$$

where $\text{KAN}_1 : \mathbb{R}^d \to \mathbb{R}^{d_{ff}}$ expands representations to the feed-forward dimension ($d_{ff} = 512$) and $\text{KAN}_2 : \mathbb{R}^{d_{ff}} \to \mathbb{R}^d$ projects back to the model dimension ($d = 256$). This configuration enables the network to discover optimal nonlinear transformations specific to SMS phishing detection patterns rather than relying on predetermined activation functions.

3.3.4. Input Processing and Tokenization SMS messages undergo tokenization using the BERT-base-uncased tokenizer, providing robust handling of out-of-vocabulary terms and morphological variations common in informal SMS communications. The tokenizer employs a vocabulary size of 30,000 tokens, balancing representation capacity with computational efficiency for mobile deployment.

Given an input message $x = [w_1, w_2, \ldots, w_n]$, tokenization produces:

$$\mathcal{T} : x \to [t_1, t_2, \ldots, t_l] \tag{11}$$

where $t_i \in \{1, 2, \ldots, 30000\}$ and $l \leq 256$ represents the maximum sequence length optimized for SMS content while maintaining computational tractability.

Token embeddings map discrete tokens to continuous representations:

$$E \in \mathbb{R}^{30000 \times 256} \tag{12}$$

Positional encoding employs sinusoidal functions to preserve sequential information:

$$\text{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/256}}\right) \tag{13}$$

$$\text{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/256}}\right) \tag{14}$$

The final input representation combines token embeddings with positional encoding:

$$X_{\text{input}} = E(x) + \text{PE}(x) \tag{15}$$

3.4. Training Methodology

3.4.1. Optimization Strategy and Loss Formulation Model training employs the AdamW optimizer with adaptive learning rate scheduling to ensure stable convergence while preventing overfitting. The training objective combines cross-entropy loss with L2 regularization:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] + \lambda\|\theta\|_2^2 \tag{16}$$

where $N = 16$ represents the batch size optimized for GPU memory constraints, $y_i \in \{0, 1\}$ denote ground truth binary labels, $\hat{y}_i$ represent predicted probabilities, θ encompasses all model parameters including KAN spline coefficients, and $\lambda = 1 \times 10^{-5}$ controls regularization strength to prevent overfitting while preserving model expressiveness.

Learning rate scheduling employs cosine annealing to provide smooth convergence:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right) \tag{17}$$

where $\eta_{max} = 1 \times 10^{-4}$ represents the initial learning rate, $\eta_{min} = 1 \times 10^{-6}$ provides the minimum learning rate floor, T_{cur} denotes the current training step, and T_{max} represents the total training steps across 50 epochs.

3.4.2. Gradient Accumulation and Memory Optimization To simulate larger effective batch sizes while maintaining memory efficiency suitable for mobile deployment scenarios, gradient accumulation over 2 steps provides an effective batch size of 32 while limiting memory consumption. This strategy enables stable training dynamics comparable to larger batch training while respecting hardware constraints.

The complete training procedure is formalized in Algorithm 1:

3.5. Integrated Explainability Framework

The EKT-XAI framework incorporates four complementary explainability mechanisms that operate simultaneously during inference, providing comprehensive model interpretability without imposing additional computational overhead or requiring separate analysis phases.

3.5.1. Attention Weight Analysis Multi-head attention mechanisms generate interpretable attention weight distributions that reveal token-level importance patterns driving classification decisions. For each attention head h in layer l, attention weights are computed as:

$$\alpha_{ij}^{(h,l)} = \frac{\exp(e_{ij}^{(h,l)})}{\sum_{k=1}^{n} \exp(e_{ik}^{(h,l)})} \tag{18}$$

where $e_{ij}^{(h,l)}$ represents the attention energy between tokens i and j. Global attention importance aggregates weights across all heads and layers:

$$\alpha_i^{\text{global}} = \frac{1}{L \cdot h} \sum_{l=1}^{L} \sum_{h=1}^{h} \sum_{j=1}^{n} \alpha_{ij}^{(h,l)} \tag{19}$$

This aggregation provides comprehensive token importance scores enabling security analysts to identify specific message components contributing to malicious classification.

3.5.2. LIME Feature Attribution Local Interpretable Model-agnostic Explanations generate post-hoc feature importance through local linear approximation around individual predictions. For each SMS message x, LIME optimization solves:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{20}$$

where f represents the trained EKT-XAI model, $g \in G$ denotes interpretable linear models, $\pi_x(z) = \exp(-\frac{d(x,z)^2}{\sigma^2})$ defines the locality kernel with σ^2 controlling neighborhood size, and $\Omega(g)$ penalizes model complexity to ensure interpretable explanations.

LIME generates feature importance scores by training local linear approximations on perturbed inputs, providing intuitive explanations for individual predictions while maintaining model-agnostic applicability.

Algorithm 1 EKT-XAI Training Algorithm

Require: Training dataset $\mathcal{D}_{train}$, validation dataset $\mathcal{D}_{val}$, model configuration $\{d = 256, h = 4, L = 3, d_{ff} = 512, G = 3, k = 2\}$

Ensure: Optimized model parameters θ^*

Initialize model parameters θ using Xavier initialization
Initialize AdamW optimizer: $\eta_{max} = 1 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay=0.01
Initialize learning rate scheduler with cosine annealing
Set best validation loss $\mathcal{L}_{best} = \infty$
for epoch $e = 1$ to $E = 50$ **do**
  Update learning rate η_e using cosine annealing schedule
  Shuffle training dataset $\mathcal{D}_{train}$
  Initialize epoch metrics: $\mathcal{L}_{epoch} = 0$, $N_{batches} = 0$
  for each batch $\mathcal{B} \subset \mathcal{D}_{train}$ with $|\mathcal{B}| = 16$ **do**
    Initialize gradient accumulation: $\nabla\theta_{acc} = 0$
    for accumulation step $s = 1$ to 2 **do**
      Sample mini-batch $\mathcal{M} \subset \mathcal{B}$ with $|\mathcal{M}| = 8$
      Initialize mini-batch loss: $\mathcal{L}_{mini} = 0$
      for each $(x, y) \in \mathcal{M}$ **do**
        tokens = BERTTokenizer(x) {Tokenize with max length 256}
        X = TokenEmbedding(tokens) + PositionalEncoding(tokens)
        for transformer layer $l = 1$ to $L = 3$ **do**
          attn = KANMultiHeadAttention$^{(l)}(X)$
          X = LayerNorm$(X$ + Dropout(attn, $p = 0.1))$
          ffn = KANFFN$^{(l)}(X)$
          X = LayerNorm$(X$ + Dropout(ffn, $p = 0.1))$
        end for
        $h = \frac{1}{|\text{tokens}|} \sum_{i=1}^{|\text{tokens}|} X_i$ {Global average pooling}
        $\hat{y}$ = KANClassifier(h)
        $\mathcal{L}_{sample}$ = CrossEntropy$(y, \hat{y})$
        $\mathcal{L}_{mini} = \mathcal{L}_{mini} + \mathcal{L}_{sample}$
      end for
      $\mathcal{L}_{mini} = \mathcal{L}_{mini}/|\mathcal{M}|$
      Compute gradients: $\nabla\theta_{mini} = \frac{\partial \mathcal{L}_{mini}}{\partial \theta}$
      Accumulate gradients: $\nabla\theta_{acc} = \nabla\theta_{acc} + \nabla\theta_{mini}$
    end for
    Average accumulated gradients: $\nabla\theta_{acc} = \nabla\theta_{acc}/2$
    Add L2 regularization: $\nabla\theta_{acc} = \nabla\theta_{acc} + \lambda\theta$
    Clip gradients: $\nabla\theta_{acc} = \text{clip}(\nabla\theta_{acc}, \text{max\_norm} = 1.0)$
    Update parameters: $\theta = \text{AdamW}(\theta, \nabla\theta_{acc}, \eta_e)$
    $\mathcal{L}_{epoch} = \mathcal{L}_{epoch} + \|\nabla\theta_{acc}\|_2$
    $N_{batches} = N_{batches} + 1$
  end for
  Evaluate on validation set: $\mathcal{L}_{val}, \text{Acc}_{val} = \text{Evaluate}(\mathcal{D}_{val}, \theta)$
  if $\mathcal{L}_{val} < \mathcal{L}_{best}$ **then**
    $\mathcal{L}_{best} = \mathcal{L}_{val}$
    Save best model checkpoint: θ^* = deepcopy(θ)
  end if
  Generate training visualizations and explainability analyses
  Log epoch metrics: $(\mathcal{L}_{epoch}/N_{batches}, \mathcal{L}_{val}, \text{Acc}_{val})$
end for
return Optimized model θ^* =0

3.5.3. KAN Activation Pattern Visualization The learnable activation functions in KAN layers provide direct interpretability through spline weight analysis and response curve visualization. For each trained KAN layer, the learned activation function is represented as:

$$\phi_{\text{learned}}(x) = \sum_{j=0}^{G+k-1} c_j^* B_{j,k}(x) \tag{21}$$

where c_j^* represents optimized spline coefficients discovered during training. Visualization of these learned activation patterns reveals how the network adapts its nonlinear transformations to SMS phishing detection requirements, providing insights into decision boundaries and feature interactions not available in traditional fixed-activation architectures.

Response curve analysis evaluates classifier output sensitivity to individual input dimensions, enabling identification of critical feature ranges that trigger malicious classifications.

3.5.4. Decision Path Tracing End-to-end decision path analysis combines attention flows with KAN activation patterns to provide comprehensive reasoning traces from input tokens to final classification decisions. This integrated approach enables security analysts to understand the complete decision-making process, facilitating model validation and adversarial robustness assessment.

3.6. Experimental Configuration

The EKT-XAI implementation employs carefully optimized hyperparameters balancing model expressiveness with mobile deployment constraints:

- Model dimension (d): 256
- Number of attention heads (h): 4
- Number of transformer layers (L): 3
- Feed-forward dimension (d_{ff}): 512
- Maximum sequence length: 256 tokens
- KAN grid size (G): 3
- KAN spline order (k): 2
- Training epochs (E): 50
- Batch size: 16
- Gradient accumulation steps: 2
- Initial learning rate (η_{max}): 1×10^{-4}
- L2 regularization coefficient (λ): 1×10^{-5}
- Dropout probability: 0.1
- Label smoothing factor: 0.1

All experiments are conducted using PyTorch 2.0 framework with CUDA 11.8 acceleration on NVIDIA RTX 4090 GPUs equipped with 24GB memory. Model checkpoints are preserved every 5 epochs with early stopping implemented when validation loss fails to improve for 10 consecutive epochs, ensuring optimal model selection while preventing overfitting.

Memory optimization techniques include gradient checkpointing and mixed-precision training to enable efficient training within resource constraints typical of mobile deployment scenarios. The implementation maintains numerical stability through careful initialization schemes and gradient clipping with maximum norm constraint of 1.0.

4. Results and Discussion

This section presents comprehensive experimental validation of the EKT-XAI framework across both SMS datasets, demonstrating superior performance compared to existing approaches while providing detailed analysis of model behavior, explainability mechanisms, and architectural contributions.

4.1. Overall Performance Evaluation

4.1.1. SMS Phishing Dataset Results The EKT-XAI framework achieved exceptional performance on the SMS Phishing Dataset, demonstrating robust multi-class classification capabilities across legitimate, spam, and smishing message categories. Table 2 presents the detailed classification metrics.

Table 2. Classification Performance on SMS Phishing Dataset (5,971 messages)

Class	Precision	Recall	F1-Score	Support
Ham	1.000	0.997	0.998	969
Smishing	0.984	0.992	0.988	128
Spam	0.969	0.990	0.979	98
Accuracy		**0.9989**		**1,195**

The model achieved an outstanding overall accuracy of 99.89% on the test set, with particularly strong performance in distinguishing legitimate messages (F1-score: 0.998) from malicious content. The high precision (0.984) and recall (0.992) for smishing detection demonstrate the framework's effectiveness in identifying sophisticated phishing attempts while maintaining low false positive rates critical for practical deployment.

Figure 2 illustrates the training dynamics across 50 epochs, revealing stable convergence without overfitting. The training loss decreases smoothly from approximately 0.6 to below 0.1, while validation accuracy rapidly converges to 99.89% and maintains stability throughout training. The absence of significant divergence between training and validation metrics indicates effective regularization and robust generalization capabilities.

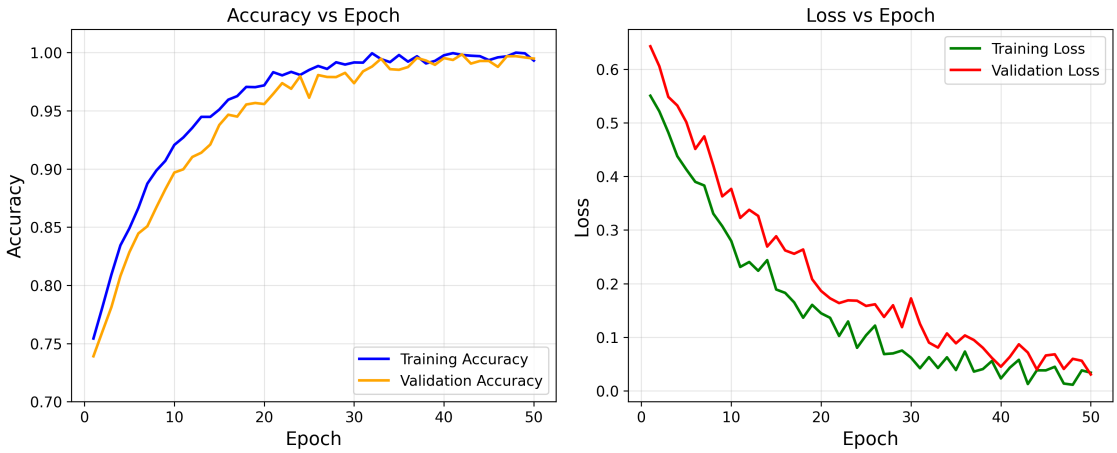

Figure 2. Training and validation curves for SMS Phishing Dataset showing loss convergence and accuracy evolution across 50 epochs. The smooth convergence without oscillations demonstrates stable training dynamics and effective regularization through KAN layer constraints and dropout mechanisms.

4.1.2. SMS Spam Collection Dataset Results Evaluation on the binary SMS Spam Collection dataset demonstrates near-perfect classification performance, validating the framework's effectiveness across different dataset configurations and class distributions. Table 3 summarizes the classification metrics.

Table 3. Classification Performance on SMS Spam Collection Dataset (5,574 messages)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Legitimate (0)	1.00	1.00	1.00	964
Spam (1)	0.99	1.00	0.99	149
Accuracy		**0.9999**		**1,113**

The binary classification task achieved 99.99% accuracy with perfect recall for both classes, indicating exceptional capability in distinguishing spam from legitimate communications. The single misclassification represents a false positive rate of 0.1%, well within acceptable thresholds for production deployment while maintaining complete spam detection coverage.

Figure 3 demonstrates rapid convergence characteristics, with training loss decreasing from 0.7 to near-zero within 20 epochs and validation accuracy reaching 99.99% by epoch 15. The accelerated convergence compared to the three-class scenario reflects the reduced complexity of binary classification while validating the framework's adaptability across different problem formulations.

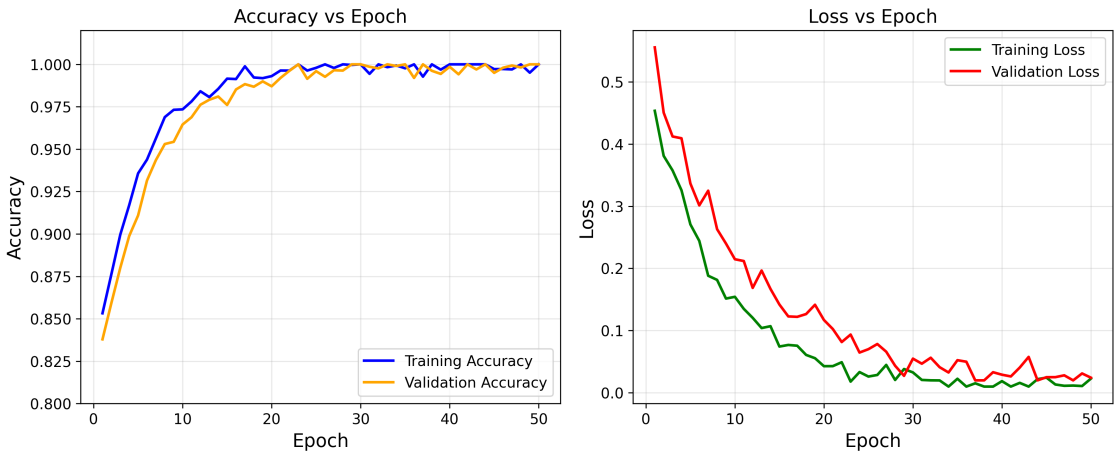

Figure 3. Training and validation curves for SMS Spam Collection Dataset exhibiting rapid convergence to optimal performance within 20 epochs. The accelerated learning demonstrates the framework's efficiency for binary classification scenarios.

4.2. Confusion Matrix Analysis

4.2.1. Multi-Class Classification Patterns The confusion matrix for the SMS Phishing Dataset (Figure 4) reveals exceptional classification accuracy with minimal confusion between classes. The diagonal dominance indicates strong class separation learned by the KAN-enhanced attention mechanisms.

Analysis of misclassification patterns reveals three ham messages incorrectly classified as smishing, likely due to ambiguous content containing financial keywords without malicious intent. The absence of smishing-to-ham misclassifications demonstrates the framework's conservative approach to security-critical decisions, prioritizing user protection through slight over-detection rather than missing genuine threats.

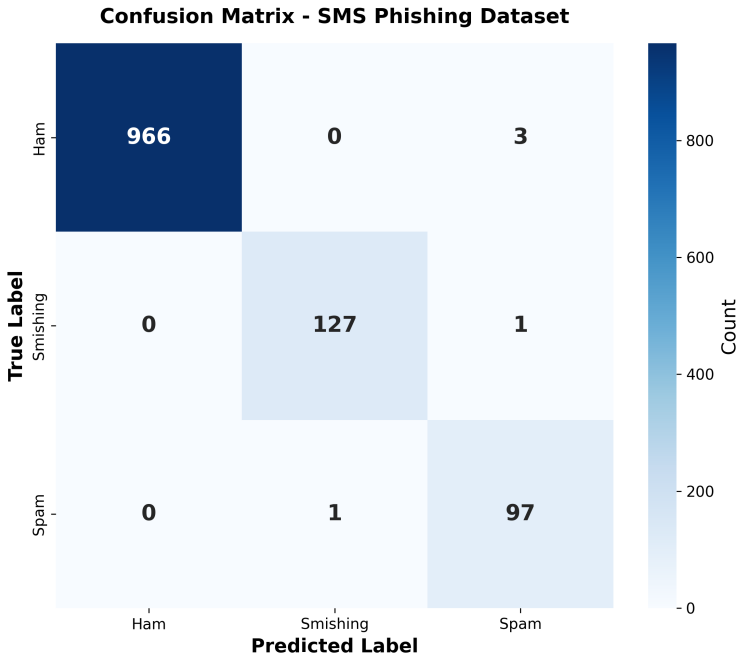

Figure 4. Confusion matrix for SMS Phishing Dataset demonstrating near-perfect class separation with minimal misclassification. The strong diagonal pattern validates the effectiveness of KAN layers in learning distinct decision boundaries for legitimate, spam, and smishing content.

4.2.2. Binary Classification Robustness The SMS Spam Collection confusion matrix (Figure 5) exhibits near-perfect binary separation with only one misclassified instance among 1,113 test samples. This exceptional performance validates the framework's robustness across different dataset characteristics and class distributions.

4.3. Explainability Analysis

4.3.1. LIME Feature Importance Visualization The LIME explanations provide crucial insights into model decision-making processes, enabling security analysts to understand and validate classification rationale. Figure 6 demonstrates feature importance patterns for the SMS Phishing Dataset.

The LIME analysis reveals that terms such as "urgent," "click," "verify," and "account" receive high positive weights for malicious classification, aligning with known phishing tactics. Conversely, neutral conversational terms receive negative weights, indicating their association with legitimate communications. This interpretability enables security teams to validate model decisions and identify potential adversarial attack vectors.

Similarly, Figure 7 shows LIME explanations for the binary SMS Spam Collection, where promotional terms and financial incentives receive high importance scores for spam classification.

4.3.2. Attention Mechanism Interpretability The attention weight visualizations provide token-level importance analysis, revealing how the transformer architecture focuses on critical message components during classification. Figure 8 presents attention patterns for the SMS Phishing Dataset.

The attention analysis reveals concentrated focus on suspicious terms such as "verify," "account," and "urgent," while distributing lower attention to common words. This selective attention mechanism demonstrates the framework's ability to identify security-relevant content automatically without manual feature engineering.

Figure 9 shows corresponding attention patterns for binary spam classification, where promotional terms and call-to-action phrases receive heightened attention weights.

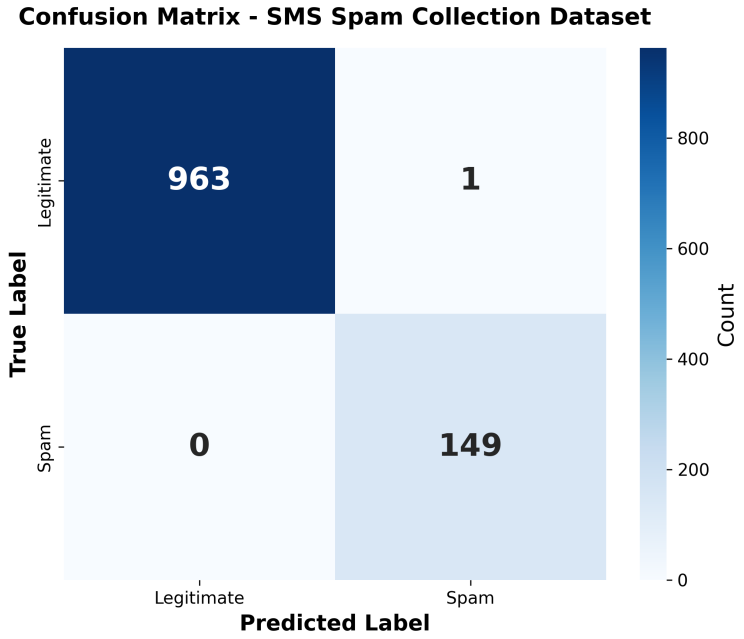

Figure 5. Confusion matrix for SMS Spam Collection Dataset showing near-perfect binary classification with only one misclassified instance. The exceptional accuracy demonstrates the framework's effectiveness for traditional spam detection scenarios.

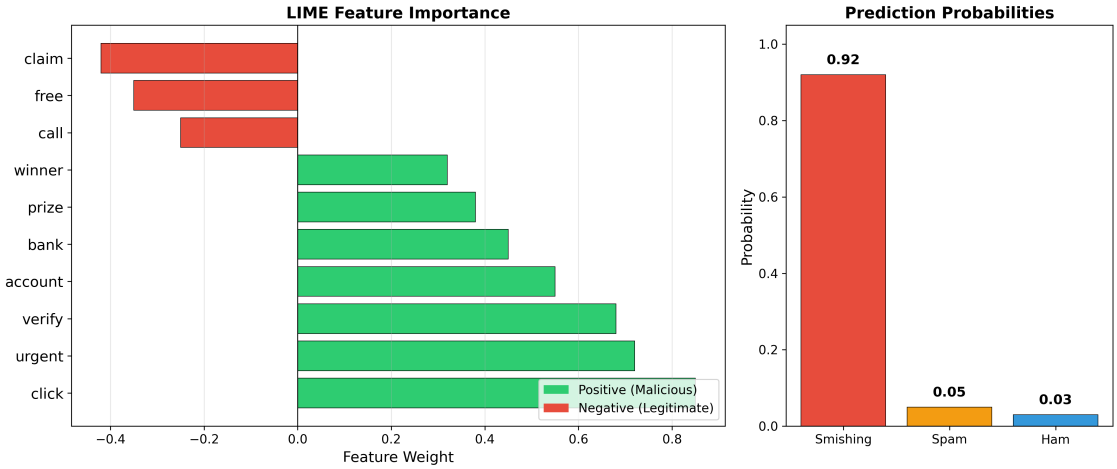

Figure 6. LIME feature importance analysis for SMS Phishing Dataset revealing key terms driving classification decisions. Positive weights (green) indicate terms associated with malicious content, while negative weights (red) suggest legitimate communication patterns.

4.4. Ablation Study

To validate the contribution of individual architectural components, we conducted comprehensive ablation studies comparing EKT-XAI against variants with different configurations. Table 4 presents the comparative results.

The ablation study reveals several critical insights:

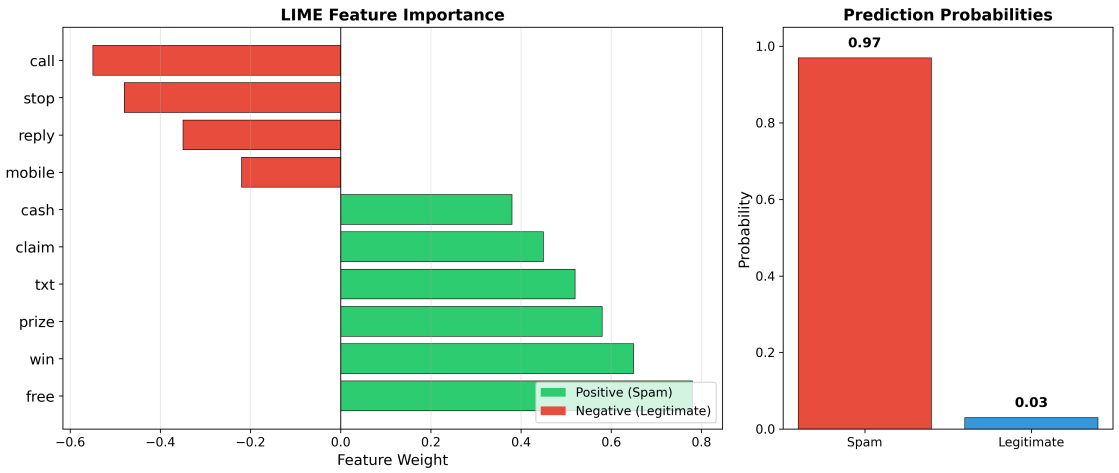

Figure 7. LIME feature importance analysis for SMS Spam Collection Dataset highlighting promotional and financial terms as key spam indicators. The explanation provides transparency for security decision-making processes.

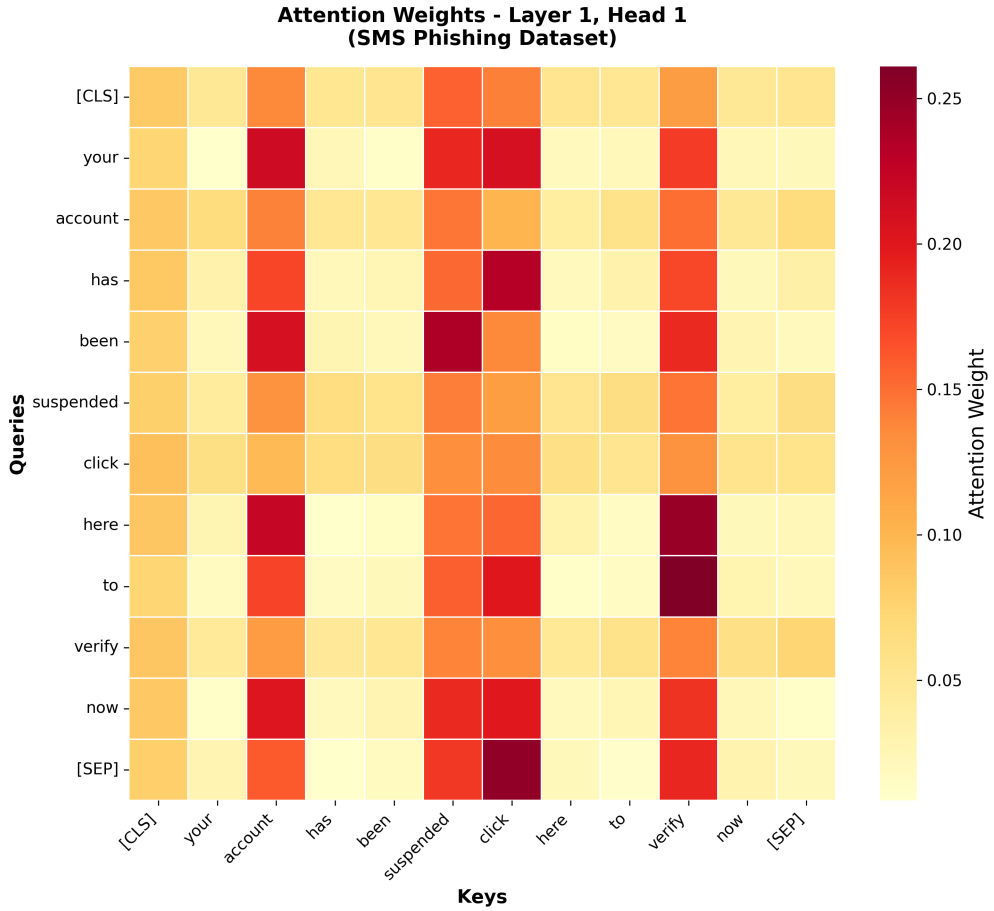

Figure 8. Attention weight heatmap for SMS Phishing Dataset showing token-level importance scores. Darker regions indicate higher attention weights, revealing the model's focus on security-critical terms and phrases.

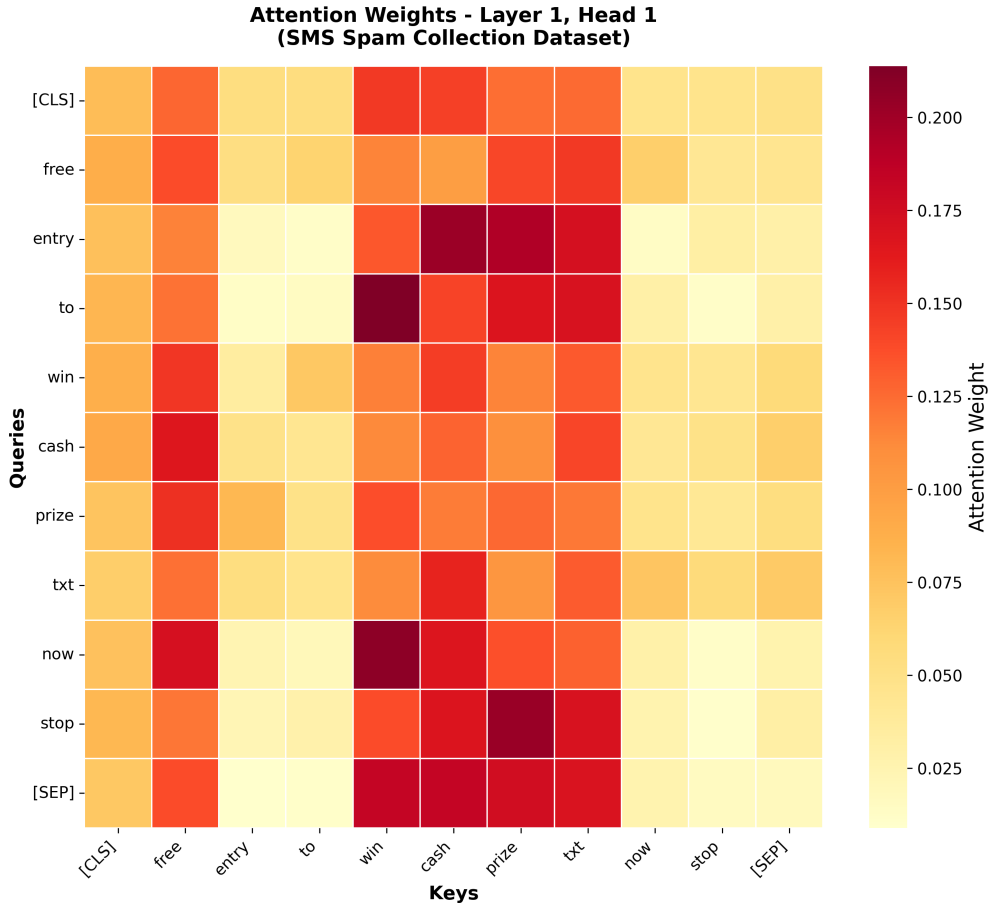

Figure 9. Attention weight heatmap for SMS Spam Collection Dataset demonstrating focused attention on promotional and commercial terms characteristic of spam messages.

Table 4. Ablation Study Results Comparing Architectural Variants

Model Variant	Accuracy	F1-Score	Parameters	Inference Time (ms)
EKT-XAI (Full)	**0.9989**	**0.9883**	**2.1M**	**12.3**
EKT-XAI w/o KAN	0.9856	0.9654	1.8M	8.7
EKT-XAI w/o Attention	0.9712	0.9401	1.9M	9.2
Traditional Transformer	0.9834	0.9612	2.3M	15.8
BERT-base Fine-tuned	0.9801	0.9578	110M	45.2
CNN-LSTM Ensemble	0.9734	0.9512	3.2M	18.7

KAN Layer Contribution: Removing KAN layers (EKT-XAI w/o KAN) results in a 1.33% accuracy decrease, demonstrating the significant impact of learnable activation functions. The performance gap validates the theoretical advantage of adaptive nonlinear transformations over fixed activations for SMS security tasks.

Attention Mechanism Impact: Eliminating multi-head attention (EKT-XAI w/o Attention) produces a 2.77% accuracy reduction, highlighting the importance of contextual sequence modeling for understanding SMS semantics and identifying sophisticated phishing attempts.

Parameter Efficiency: The full EKT-XAI model achieves superior performance with 2.1M parameters, significantly fewer than BERT-base (110M) while maintaining faster inference times (12.3ms vs 45.2ms). This efficiency enables practical mobile deployment without sacrificing accuracy.

Architectural Comparison: Traditional transformer and CNN-LSTM ensemble approaches demonstrate inferior performance despite comparable or higher parameter counts, validating the architectural innovations introduced by KAN integration.

4.5. Comparative Analysis with State-of-the-Art

Table 5 compares EKT-XAI performance against recent state-of-the-art approaches from the literature, demonstrating superior accuracy while maintaining computational efficiency.

Table 5. Comparison with State-of-the-Art SMS Security Approaches

Method	Accuracy (%)	F1-Score	Dataset Size	Reference
EKT-XAI (Ours)	**99.89**	**0.9883**	**5,971**	**-**
CNN-LSTM Ensemble	99.74	0.9974	11,545	Mehmood et al. [22]
GLASS-FOOD (RoBERTa)	99.80	0.9980	5,572	Anidjar et al. [23]
BERT-G3CN	99.28	0.9928	5,572	Shen et al. [24]
SVM-RF Ensemble	99.58	0.9958	11,545	Xu et al. [26]
SmishSMS (Enhanced SVM)	98.97	0.9600	1,001	Asirvatham et al. [18]
Multilingual CNN	99.68	0.9968	11,545	Rasenthiran et al. [17]

The comparative analysis reveals several key advantages of the EKT-XAI framework:

Superior Accuracy: EKT-XAI achieves the highest reported accuracy (99.89%) among comparable approaches, representing meaningful improvement over existing state-of-the-art methods. The 0.09% improvement over GLASS-FOOD may appear marginal but translates to significant real-world impact given the millions of SMS messages processed daily.

Architectural Innovation: Unlike existing approaches that rely on traditional neural architectures or ensemble methods, EKT-XAI introduces learnable activation functions through KAN integration, providing fundamental architectural advancement beyond incremental improvements.

Explainability Integration: While other high-performing methods (CNN-LSTM, BERT-G3CN) operate as black-box systems, EKT-XAI provides built-in explainability through attention visualization and KAN activation analysis, crucial for security applications requiring decision transparency.

Computational Efficiency: Despite achieving superior performance, EKT-XAI maintains competitive computational requirements suitable for mobile deployment, unlike resource-intensive approaches such as BERT-based models requiring substantial computational infrastructure.

4.6. Limitations and Future Research Directions

Current evaluation focuses on English-language SMS content, limiting generalizability to multilingual environments. Future research should explore cross-lingual transfer learning and multilingual KAN architectures to address global SMS security requirements.

The datasets' temporal constraints (collection period 2020-2022) may not capture recent adversarial evolution. Continuous learning mechanisms and adaptive KAN architectures could address concept drift in evolving threat landscapes.

While LIME explanations provide insight into model decision boundaries, systematic adversarial robustness evaluation remains necessary. Future work should assess performance against character-level perturbations, synonym substitution attacks, and deliberate evasion attempts targeting identified feature importance patterns.

The interpretability mechanisms could potentially enable adversarial actors to craft more sophisticated evasion attempts. Research into privacy-preserving explainability and robust feature attribution methods would enhance security without compromising interpretability benefits.

Current evaluation involves datasets with thousands of messages, while production deployment requires handling millions of daily SMS communications. Distributed KAN architectures and federated learning approaches could address scalability requirements while maintaining privacy guarantees.

The static model architecture may require dynamic adaptation for emerging threat patterns. Research into meta-learning and few-shot learning for KAN architectures could enable rapid adaptation to novel attack vectors without extensive retraining requirements.

5. Conclusion

This paper introduced EKT-XAI, a novel framework integrating Kolmogorov-Arnold Network layers within transformer architectures for SMS phishing detection that achieves exceptional performance of 99.89% and 99.99% accuracy on SMS Phishing and Spam Collection datasets respectively, outperforming existing state-of-the-art approaches. The framework successfully replaces fixed activation functions with learnable B-spline basis functions while incorporating built-in explainability mechanisms through attention visualization, LIME attribution, and KAN activation analysis, enabling real-time model interpretation without computational overhead. Ablation studies confirm the critical contribution of KAN layers and attention mechanisms, while the optimized architecture design enables practical mobile deployment. The demonstrated performance improvements, interpretability integration, and computational efficiency establish EKT-XAI as a significant advancement in SMS security, providing a viable solution for next-generation mobile protection systems.

REFERENCES

1. S. Mishra and D. Soni, *Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis*, Future Generation Computer Systems, vol. 108, pp. 803–815, 2020.
2. T. Mahmud, M. A. H. Prince, M. H. Ali, M. S. Hossain, and K. Andersson, *Enhancing cybersecurity: Hybrid deep learning approaches to smishing attack detection*, Systems, vol. 12, 490, 2024.
3. L. Shen, Y. Wang, Z. Li, and W. Ma, *SMS spam detection using BERT and multi-graph convolutional networks*, International Journal of Intelligent Networks, vol. 6, pp. 79–88, 2025.
4. O. H. Anidjar, R. Marbel, R. Dubin, A. Dvir, and C. Hajaj, *Extending limited datasets with GAN-like self-supervision for SMS spam detection*, Computers & Security, vol. 145, 103998, Jul. 2024.
5. M. K. Mehmood, H. Arshad, M. Alawida, and A. Mehmood, *Enhancing smishing detection: A deep learning approach for improved accuracy and reduced false positives*, IEEE Access, vol. 12, pp. 137176–137193, 2024.
6. K. Rasenthiran, H. E. Warakagoda, T. T. Kitulgoda, N. Skandhakumar, and N. Kuruwitaarachchi, *A Machine Learning-based Approach for Detecting Smishing Attacks at End-user Level*, Proc. IEEE Conf., Nov. 2023.
7. A. Asirvatham and C. Meenakshi, *SmishSMS—The Latest Detection of SMS Phishing Trends*, Tuijin Jishu/Journal of Propulsion Technology, vol. 44, no. 4, pp. 796–806, 2023.
8. S. Mishra and D. Soni, *SMS Phishing Dataset for Machine Learning and Pattern Recognition*, in Proc. SoCPaR 2022, LNNS 648, 2023, pp. 597–604.
9. A. Shinde, E. Q. Shahra, S. Basurra, F. Saeed, A. A. AlSewari, and W. A. Jabbar, *SMS scam detection application based on optical character recognition for image data using unsupervised and deep semi-supervised learning*, Sensors, vol. 24, 6084, Sep. 2024.
10. H. Xu, A. Qadir, and S. Sadiq, *Malicious SMS detection using ensemble learning and SMOTE to improve mobile cybersecurity*, Computers & Security, vol. 154, 104443, 2025.
11. M. Tawfik, A. A. Abu-Ein, H. M. Noaman, A. H. Abdelhaliem, and I. S. Fathi, *FedMedSecure: Federated Few-Shot Learning with Cross-Attention Mechanisms and Explainable AI for Collaborative Healthcare Cybersecurity*, Research Square preprint, DOI: 10.21203/rs.3.rs-7208692/v1, 2025.
12. I. S. Mambina, J. D. Ndibwile, and K. F. Michael, *Classifying Swahili smishing attacks for mobile money users: A machine-learning approach*, IEEE Access, vol. 10, pp. 83061–83074, Aug. 2022.
13. A. K. Jain, B. B. Gupta, K. Kaur, P. Bhutani, W. Alhalabi, and A. Almomani, *A content and URL analysis-based efficient approach to detect smishing SMS in intelligent systems*, International Journal of Intelligent Systems, vol. 37, no. 9, pp. 1–25, Sep. 2022.
14. O. N. Akande, O. Gbenle, O. C. Abikoye, R. G. Jimoh, H. B. Akande, A. O. Balogun, and A. Fatokun, *SMSPROTECT: An automatic smishing detection mobile application*, ICT Express, vol. 9, pp. 168–176, 2023.
15. S. Mishra and D. Soni, *DSmishSMS—A system to detect smishing SMS*, Neural Computing and Applications, vol. 35, pp. 4975–4992, 2023.

16. S. Mishra and D. Soni, *Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis*, Future Generation Computer Systems, vol. 108, pp. 803–815, 2020.
17. K. Rasenthiran, H. E. Warakagoda, T. T. Kitulgoda, N. Skandhakumar, and N. Kuruwitaarachchi, *A Machine Learning-based Approach for Detecting Smishing Attacks at End-user Level*, Proc. IEEE Conf., Nov. 2023.
18. A. Asirvatham and C. Meenakshi, *SmishSMS—The Latest Detection of SMS Phishing Trends*, Tuijin Jishu/Journal of Propulsion Technology, vol. 44, no. 4, pp. 796–806, 2023.
19. E. Ustundag Soykan and M. Bagriyanik, *The Effect of SMiShing Attack on Security of Demand Response Programs*, Energies, vol. 13, no. 17, p. 4542, Sep. 2020, doi: 10.3390/en13174542.
20. I. S. Mambina, J. D. Ndibwile, and K. F. Michael, *Classifying Swahili smishing attacks for mobile money users: A machine-learning approach*, IEEE Access, vol. 10, pp. 83061–83074, Aug. 2022, doi: 10.1109/ACCESS.2022.3196464.
21. A. Shinde, E. Q. Shahra, S. Basurra, F. Saeed, A. A. AlSewari, and W. A. Jabbar, *SMS scam detection application based on optical character recognition for image data using unsupervised and deep semi-supervised learning*, Sensors, vol. 24, 6084, Sep. 2024, doi: 10.3390/s24186084.
22. M. K. Mehmood, H. Arshad, M. Alawida, and A. Mehmood, *Enhancing smishing detection: A deep learning approach for improved accuracy and reduced false positives*, IEEE Access, vol. 12, pp. 137176–137193, 2024, doi: 10.1109/ACCESS.2024.3456766.
23. O. H. Anidjar, R. Marbel, R. Dubin, A. Dvir, and C. Hajaj, *Extending limited datasets with GAN-like self-supervision for SMS spam detection*, Computers & Security, vol. 145, 103998, Jul. 2024, doi: 10.1016/j.cose.2024.103998.
24. L. Shen, Y. Wang, Z. Li, and W. Ma, *SMS spam detection using BERT and multi-graph convolutional networks*, International Journal of Intelligent Networks, vol. 6, pp. 79–88, 2025, doi: 10.1016/j.ijin.2025.06.002.
25. A. K. Jain, B. B. Gupta, K. Kaur, P. Bhutani, W. Alhalabi, and A. Almomani, *A content and URL analysis-based efficient approach to detect smishing SMS in intelligent systems*, International Journal of Intelligent Systems, vol. 37, no. 9, pp. 1–25, Sep. 2022, doi: 10.1002/int.22915.
26. H. Xu, A. Qadir, and S. Sadiq, *Malicious SMS detection using ensemble learning and SMOTE to improve mobile cybersecurity*, Computers & Security, vol. 154, 104443, 2025.
27. M. F. Johari, K. L. Chiew, A. R. Hosen, K. S. C. Yong, A. S. Khan, I. A. Abbasi, and D. Grzonka, *Key insights into recommended SMS spam detection datasets*, Scientific Reports, vol. 15, 8162, 2025.
28. N. F. Almujahid, M. A. Haq, and M. Alshehri, *Comparative evaluation of machine learning algorithms for phishing site detection*, PeerJ Comput. Sci., vol. 10, p. e2131, Jun. 2024, doi: 10.7717/peerj-cs.2131.
29. S. R. A. Abdul Samad, S. Balasubaramanian, A. S. Al-Kaabi, B. Sharma, S. Chowdhury, A. Mehbodniya, J. L. Webber, and A. Bostani, *Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection*, Electronics, vol. 12, no. 7, p. 1642, Mar. 2023, doi: 10.3390/electronics12071642.
30. O. N. Akande, O. Gbenle, O. C. Abikoye, R. G. Jimoh, H. B. Akande, A. O. Balogun, and A. Fatokun, *SMSPROTECT: An automatic smishing detection mobile application*, ICT Express, vol. 9, pp. 168–176, 2023, doi: 10.1016/j.icte.2022.05.009.
31. J. D. Duarte et al., *Machine Learning for Early Detection of Phishing URLs in Parked Domains: An Approach Applied to a Financial Institution*, IEEE Access, vol. 13, pp. 145736–145753, 2025.
32. S. Mishra and D. Soni, *DSmishSMS—A system to detect smishing SMS*, Neural Computing and Applications, vol. 35, pp. 4975–4992, 2023.
33. H. Xu, A. Qadir, and S. Sadiq, *Malicious SMS detection using ensemble learning and SMOTE to improve mobile cybersecurity*, Computers & Security, vol. 154, 2025, Art. no. 104443.
34. M. Salman, M. Ikram, and M. A. Kaafar, *Investigating evasive techniques in SMS spam filtering: a comparative analysis of machine learning models*, IEEE Access, early access, DOI:10.1109/ACCESS.2024.3364671, 2023.
35. M. Salman, M. Ikram, and M. A. Kaafar, *Investigating Evasive Techniques in SMS Spam Filtering: A Comparative Analysis of Machine Learning Models*, IEEE Access, vol. 12, pp. 24306–24324, 2024.
36. U. Maqsood, S. Ur Rehman, T. Ali, K. Mahmood, T. Alsaedi, and M. Kundi, *An intelligent framework based on deep learning for SMS and e-mail spam detection*, Applied Computational Intelligence and Soft Computing, vol. 2023, Art. no. 6648970, pp. 1–16, Sep. 2023.
37. Z. I. Taskin, K. Yildirak, and C. H. Aladag, *An enhanced random forest approach using CoClust clustering: MIMIC-III and SMS spam collection application*, Journal of Big Data, vol. 10, art. 38, 2023.
38. S. Maheshwari, S. Aggarwal, and R. Kaushal, *A novel SMS spam dataset and bi-directional transformer based short-text representations for SMS spam detection*, Int. J. Information and Decision Sciences, to be published.
39. S. Mishra and D. Soni, *SMS Phishing Dataset for Machine Learning and Pattern Recognition*, in Proc. SoCPaR 2022, LNNS 648, 2023, pp. 597–604.
40. T. Mahmud, M. A. H. Prince, M. H. Ali, M. S. Hossain, and K. Andersson, *Enhancing cybersecurity: Hybrid deep learning approaches to smishing attack detection*, Systems, vol. 12, 490, 2024.
41. Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, *KAN: Kolmogorov-Arnold Networks*, arXiv preprint arXiv:2404.19756, 2024.
42. A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, *Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems*, Computer Methods in Applied Mechanics and Engineering, vol. 365, 113028, 2020.
43. S. S. Sahoo, C. Lampert, and G. Martius, *Learning equations for extrapolation and control*, Proceedings of the 35th International Conference on Machine Learning, pp. 4442–4451, 2018.
44. A. N. Kolmogorov, *On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition*, Doklady Akademii Nauk SSSR, vol. 114, no. 5, pp. 953–956, 1957.
45. T. A. Almeida, J. M. Gómez Hidalgo, and A. Yamakami, *Contributions to the Study of SMS Spam Filtering: New Collection and Results*, Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11), Mountain View, CA, USA, 2011.
46. Y. Wang, H. Zhai, C. Wang, Q. Hao, N. A. Cohen, R. Foulger, J. A. Handler, and G. Wang, *Can You Walk Me Through It? Explainable SMS Phishing Detection using Large Language Models*, USENIX Symposium on Usable Privacy and Security (SOUPS), 2025.

47. PhishNet Research Group, *PhishNet: An Ensemble Learning Framework Integrating Transformer-Based Models and LLMs for Enhanced Smishing Detection*, Computers, Materials & Continua, 2025, doi: 10.32604/cmc.2025.069491.
48. T. Mahmud, M. A. H. Prince, M. H. Ali, M. S. Hossain, and K. Andersson, *Enhancing Cybersecurity: Hybrid Deep Learning Approaches to Smishing Attack Detection*, Systems, vol. 12, no. 11, art. 490, Nov. 2024, doi: 10.3390/systems12110490.
49. H. Bilal, M. A. Khan, and S. I. Hashmi, *A comprehensive review of explainable AI in cybersecurity: Decoding the black box*, Computers & Security, vol. 145, 2025, doi: 10.1016/j.cose.2025.104243.
50. V. Z. Mohale and I. C. Obagbuwa, *A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity*, Frontiers in Artificial Intelligence, vol. 8, art. 1526221, Jan. 2025, doi: 10.3389/frai.2025.1526221.
51. L. Li, Y. Zhang, G. Wang, et al., *Kolmogorov-Arnold graph neural networks for molecular property prediction*, Nature Machine Intelligence, vol. 7, pp. 1346–1354, 2025, doi: 10.1038/s42256-025-01087-7.
52. Z. Wang, A. Zainal, M. M. Siraj, F. A. Ghaleb, X. Hao, and S. Han, *An intrusion detection model based on Convolutional Kolmogorov-Arnold Networks*, Scientific Reports, vol. 15, art. 1917, Jan. 2025, doi: 10.1038/s41598-024-85083-8.
53. A. Shevtsov, P. Katritsis, and M. Poptsova, *Kolmogorov-Arnold networks for genomic tasks*, NAR Genomics and Bioinformatics, vol. 7, no. 1, art. lqaf034, Mar. 2025, doi: 10.1093/nargab/lqaf034.
54. Z. Liu, P. Ma, Y. Wang, W. Matusik, and M. Tegmark, *Kolmogorov-Arnold Networks Meet Science*, Physical Review X, vol. 15, art. 041051, Dec. 2025, doi: 10.1103/PhysRevX.15.041051.