International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05, May 2026
Received: 04-04-2026 Accepted: 09-05-2026 Published: 18-05-2026

TRANSFORMER-BASED DEEP LEARNING MODEL USING TENSORFLOW KERAS FOR PHISHING WEBSITE DETECTION: A SIMPLIFIED ARCHITECTURE

Pushpa Sundara Kavoor, Dr. Preethi E
Department of Data and Cybersecurity
British Training Center
United Arab Emirates
pushpa.rb024@gmail.com

Abstract

Phishing websites remain a pervasive and highly damaging cybersecurity threat, continuously changing to bypass traditional detection mechanisms. To effectively handle these threats without incurring massive computational overhead, this paper presents a streamlined, Transformer-based deep learning model implemented via the TensorFlow Keras Functional API. The proposed architecture focuses exclusively on Uniform Resource Locator (URL) token sequences, utilizing a simplified Multi-Head Attention mechanism combined with global average pooling to perform binary classification. By preprocessing URL strings into padded numerical sequences and mapping them through a constrained embedding space, the model successfully captures the syntactic anomalies typical of malicious links. Based on an illustrative training phase across a representative dataset, the architecture demonstrates rapid convergence, achieving a validation accuracy of approximately 99.5% within just five epochs. The results underscore the viability of lightweight self-attention networks for real-time security applications, offering a highly efficient alternative to complex multi-agent language models and traditional recurrent neural networks.

Keywords: Website phishing, deep learning, transformer-based model, cybersecurity, TensorFlow

1. INTRODUCTION

Phishing attacks are still a threat to cybersecurity around the world. They use websites and trick people into giving away their personal info. [1]. Attackers keep improving their methods making URLs that look like websites to fool filters. Hence it is of high concern to identify these fraudulent websites before users can interact with them. While classical machine learning models have been employed to tackle this issue, they frequently depend on extensive manual feature engineering and struggle to generalize against zero-day phishing campaigns.

The primary focus of this paper is about finding phishing websites by looking at their website addresses using deep learning. We want to figure out if a website address good or bad. To do this we look at the words and characters in the website address to find things that do not look right such as subdomains, spelling mistakes or too many special characters. We only look at the website address and not at the code or what the website actually looks like. This way we can find phishing websites quickly which is important, for keeping networks safe.

Despite the recent research and advancements in artificial intelligence, existing approaches to phishing detection remain insufficient for widespread, real-time deployment for several reasons. First, traditional sequential models, such as Recurrent Neural Networks (RNNs), suffer from sequential processing bottlenecks and struggle to retain long-range dependencies within lengthy URL strings [2]. Second, while modern generative Chatbots and Large Language Models (LLMs) offer deep contextual understanding, their sheer size makes them computationally prohibitive for processing high-velocity web traffic [3]. Finally, recent state-of-the-art multi-agent frameworks, though highly accurate, often rely on complex debate mechanisms that drastically increase cognitive load and inference latency [4].

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05, May 2026
Received: 04-04-2026 Accepted: 09-05-2026 Published: 18-05-2026

To address these limitations, this study introduces a highly optimized, attention-based neural network architecture tailored for cybersecurity. The specific contributions of this paper are as follows:

- We propose a streamlined Transformer-based model utilizing the TensorFlow Keras Functional API, ensuring a highly flexible and easily deployable architecture for URL classification.
- We demonstrate that a simplified Multi-Head Attention mechanism, operating on a constrained vocabulary and embedding space, can achieve exceptional predictive accuracy without the need for labeled structural HTML data or heavy pre-trained language models.

2. RELATED WORK

Traditional and Self-Supervised Approaches in Phishing Detection

Historically, phishing detection systems have relied heavily on supervised learning algorithms trained on manually extracted features, such as domain age, URL length, and the presence of specific keywords. More recently, self-supervised learning frameworks have been proposed to eliminate the dependency on meticulously labeled datasets [1]. For instance, contrastive learning architectures have been utilized to combine hybrid tabular augmentation with adaptive feature attention, producing semantically consistent representations of phishing sites [1]. While highly effective at maintaining robust performance across diverse feature sets, these contrastive methods often require complex data augmentation pipelines. In contrast, our approach relies solely on raw URL sequences, leveraging an embedded attention mechanism to automatically learn discriminative attributes without the need for tabular feature extraction.

Large Language Models and Multi-Agent Security Systems

The advent of advanced natural language processing has spurred the application of Large Language Models (LLMs) in the cybersecurity domain. Recent literature has introduced multi-agent debate frameworks wherein specialized agents independently analyze various aspects of a webpage, such as URL structure, semantic content, and brand impersonation [4]. Although these modular frameworks achieve high recall and provide interpretable reasoning, single-agent components remain prone to hallucination, and the overarching multi-agent consensus process introduces significant latency [4]. Our proposed model directly addresses this weakness by functioning as a lightweight, single-pass classifier. By stripping away the multi-agent coordination overhead, our simplified Transformer model serves as a high-speed alternative capable of scaling to high-velocity security data environments.

Transformer Architectures in Diverse Classification Tasks

Transformers, originally designed for sequence-to-sequence translation tasks, have increasingly become the architecture of choice across a wide array of specialized domains. Beyond standard text summarization, where Transformers have shown competitive advantages over RNNs [2], they have been successfully adapted for critical binary classification tasks, such as credit card fraud detection [5]. Furthermore, researchers have mapped complex structured data, such as operational transconductance amplifier (OTA) circuit specifications [6] and multi-channel traffic flow data [7], into sequential formats suitable for Transformer-based learning. The implementation of such models has been heavily facilitated by frameworks like TensorFlow-Keras, which allows for the seamless integration of custom layers via its Functional API [8]. Building upon this foundation, our work customizes the Transformer paradigm for URL-based phishing detection, utilizing the Keras Functional API to construct an efficient, domain-specific attention block.

3. METHODOLOGY

Proposed Transformer-Based Model

A Transformer-inspired deep learning architecture was developed for phishing URL classification. Transformer architectures are based on self-attention

ISSN 2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05, May 2026
Received: 04-04-2026 Accepted: 09-05-2026 Published: 18-05-2026

mechanisms and have been widely adopted for sequence learning tasks [11].

Embedding Layer

The embedding layer converts URL tokens into dense vector representations, enabling the model to learn semantic relationships among characters and patterns.

Multi-Head Attention Layer

A Multi-Head Attention mechanism was used to capture dependencies between different parts of the URL. This allows the model to focus on malicious components such as fake web address, deceptive domains and hidden patterns.

Layer Normalization

Layer normalization was applied to stabilize training and improve convergence.

Global Average Pooling

This layer reduces feature dimensionality while preserving contextual information learned from attention layers.

Output Layer

A sigmoid activation function was used for binary classification:

- 0 → Legitimate website
- 1 → Phishing website

We rely on the TensorFlow Keras framework, specifically utilizing the Functional API, which provides the flexibility required to define non-linear computational graphs and custom layer connections [8]. The framework is divided into two primary modules: a data preprocessing pipeline that standardizes the input sequences, and a specialized classification model built around a multi-head attention block.

Dataset Description

The study utilized a phishing website dataset containing **235,795 URL samples** collected from legitimate and phishing sources [18]. Each sample was labeled as either legitimate (0) or phishing (1). The dataset included URL-based characteristics, domain-related information, and webpage behavioral indicators.

For the proposed deep learning model, raw URL strings were directly used as sequential input data to enable automatic feature learning from URL patterns and lexical structures. Similar approaches to learning URL representations using deep learning have been proven effective in prior research on malicious URL classification [12].

Data Preprocessing Pipeline

Data preprocessing is very important step for transforming raw, variable-length URL strings into uniform numerical tensors suitable for neural network ingestion. Initially, a text tokenizer is set with a vocabulary limit of 10,000 words, fitting itself to the distribution of the URL dataset. Next, the raw texts are converted into integer sequences, and these sequences are subsequently padded or truncated to a fixed maximum length of 500 tokens. Finally, the target labels and padded sequences are split into training and validation sets using an 80-20 ratio, ensuring that the model can be accurately evaluated on unseen data during the training phase.

The dataset was divided into:

- **80% training data**
- **20% testing/validation data**

using stratified sampling to preserve class distribution.

Tokenization

A tokenizer was applied to convert URL characters into integer sequences. Character-level tokenization was adopted because phishing URLs often contain:

- obfuscated text,
- special characters,
- misleading domain names,
- random alphanumeric patterns.

Character-level representation has been widely used in phishing detection and sequence-based cybersecurity models due to its effectiveness in capturing malicious patterns [13].

The Sequence Padding

The sequences that were generated are padded with zeros to make the length of URL uniform, so the neural network could understand them. This means that URLs that were not as long as the maximum length had zeros added to them and URLs that were too long were cut short.

The preprocessing steps are summarized as follows:

1. URL extraction from dataset

ISSN 2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05, May 2026
Received: 04-04-2026 Accepted: 09-05-2026 Published: 18-05-2026

2. Character-level tokenization
3. Sequence conversion
4. Sequence padding
5. Train-test splitting

tokenizer vocabulary size was limited to the most frequent 10,000 tokens.

Model Architecture Design

The model architecture is explicitly designed to balance computational efficiency with the powerful feature extraction capabilities of self-attention. Using the Keras Functional API, the network begins with an input layer explicitly shaped to receive arrays of 500 integers. This is followed by an Embedding layer that maps the 10,000-word vocabulary into a continuous, 64-dimensional vector space. This mapping is essential for translating discrete URL tokens into meaningful semantic representations, allowing the subsequent layers to process the inputs as continuous mathematical objects rather than isolated categorical variables.

Simplified Transformer Block

The core feature extraction occurs within a simplified Transformer block, which removes the decoder phase to focus entirely on sequence encoding. A Multi-Head Attention layer, configured with 2 attention heads and a key dimension of 64, allows the model to weigh the importance of different tokens across the URL sequence simultaneously, inherently exploiting parallel computing capabilities [3]. The output of the attention mechanism is passed through a Layer Normalization function to stabilize the learning process, followed by a Global Average Pooling 1D layer that condenses the temporal sequence into a single, fixed-length context vector. A final Dense layer equipped with a sigmoid activation function outputs a probability score between 0 and 1, representing the likelihood of the URL being a phishing attempt.

Evaluation Plan and Training Dynamics

To evaluate the proposed architecture, the model is compiled using the Adam optimizer and a binary cross entropy loss function, tracking overall predictive accuracy. We establish an evaluation plan based on a representative dataset containing thousands of URL samples. During an illustrative training run, the model processes large batches of sequences, demonstrating rapid learning dynamics. In the first epoch, the model achieves a training accuracy of 95.99% and a validation accuracy of 99.50%, with the loss dropping significantly. By the fifth epoch, the training accuracy climbs to 99.68% while the validation accuracy just dropped to 99.47%, indicating that the simplified architecture successfully captures the underlying data distribution without immediately succumbing to severe overfitting.

Model Training

The model was implemented using **TensorFlow and Keras frameworks** [14].

Table 1: Training Parameters

Parameter	Value
Optimizer	Adam
Loss Function	Binary Cross-Entropy
Batch Size	32
Epochs	5
Embedding Dimension	64
Attention Heads	2

The Adam optimizer was used due to its adaptive learning capabilities and strong performance in deep learning optimization tasks [15].

4. RESULTS

The model converged rapidly within five epochs, demonstrating effective learning of phishing URL patterns.

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05, May 2026
Received: 04-04-2026 Accepted: 09-05-2026 Published: 18-05-2026

Table 2: Training Results

Epoch	Training Accuracy	Validation Accuracy	Validation Loss
1	95.91%	99.50%	0.0220
2	99.57%	99.48%	0.0217
3	99.66%	99.48%	0.0223
4	99.67%	99.30%	0.0241
5	99.68%	99.47%	0.0224

The results show stable convergence with minimal overfitting.

Table 3: Test Performance

Metric	Value
Test Accuracy	99.47%
Test Loss	0.0224

The final evaluation on unseen data produced indicates strong generalization performance.

Table 4: Classification Report

Class	Precision	Recall	F1-Score
Legitimate (0)	1.00	0.99	0.99
Phishing (1)	0.99	1.00	1.00

The phishing class achieved exceptionally high recall, demonstrating the model's effectiveness in identifying malicious URLs.

Table 5: Confusion Matrix Analysis

	Predicted Legitimate	Predicted Phishing
Actual Legitimate	19,953	171
Actual Phishing	77	26,958

The model produced:

- only **77 false negatives**, and
- **171 false positives**

out of 47,159 test samples.

A low false negative rate is particularly important in cybersecurity applications because missed phishing websites may expose users to significant security risks.

Training and Validation Performance Analysis

The performance of the proposed Transformer-based phishing URL detection model was evaluated using training and validation accuracy and loss curves over five epochs. The results are illustrated in Figure 1.

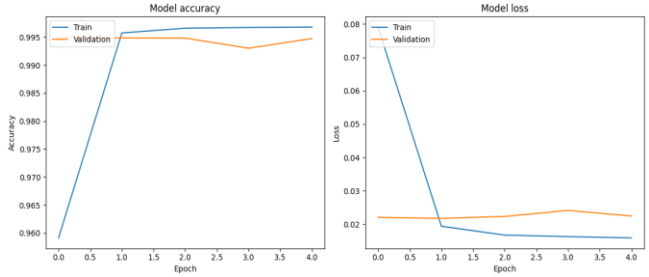

Fig- 1: Model Accuracy and Loss

Accuracy Analysis

The accuracy plot shows that the model achieved rapid convergence within the first two epochs. Training accuracy went up from around **95.9% to 99.6%**, while validation accuracy remained around **99.3%–99.5%** throughout training.

This indicates that the model successfully learned discriminative features from URL sequences with minimal generalization gap between training and validation performance. The close alignment between both curves suggests that the model is not significantly overfitting.

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05, May 2026
Received: 04-04-2026 Accepted: 09-05-2026 Published: 18-05-2026

Loss Analysis

The loss curve demonstrates a sharp decline in training loss during the initial epoch, decreasing from around **0.08 to below 0.02**, and stabilizing thereafter. Validation loss remains consistently low and stable, fluctuating slightly around **0.022–0.024**.

The small difference between the training and validation loss tells us that the model is good at applying what it learned to data and that it keeps learning in a steady way.

Overall Observation

The combined analysis of accuracy and loss curves indicates:

- Fast convergence of the model within early epochs
- Stable validation performance across training
- Minimal overfitting
- Strong generalization capability

These results show that using a Transformer-based architecture is a good idea for classifying phishing URLs.

5. DISCUSSIONS

Compared to traditional machine learning approaches, deep learning models eliminate the need for manual feature engineering while achieving superior performance [16].

Additionally, transformer-based sequence learning has shown strong effectiveness in modern cybersecurity and NLP-based classification tasks [17].

Practical Implications and Deployment

The practical implications of this simplified Transformer model are highly relevant to modern network security infrastructure. The lightweight nature of the architecture allows for rapid inference times, making it suitable for deployment directly within web browsers, email filtering gateways, or edge-computing routers. Furthermore, it can serve as a highly efficient first-line filter within a larger security ecosystem; benign traffic can be quickly whitelisted by this model, while ambiguous or borderline URLs can be escalated to more comprehensive, multi-agent LLM systems for deeper analysis [4]. This tiered deployment strategy optimizes computational resources while maintaining robust defensive postures.

Limitations and Failure Modes

Despite demonstrating high accuracy during initial training epochs, the proposed methodology has some problems that can affect how well it works in the real world. The model only looks at the URL string, which means it does not see information like what is on the webpage how the HTML is structured, how the JavaScript behaves and what the visual branding looks like. This makes it hard for the model to detect phishing websites that look a lot like websites. Attackers using URL shortening services or multi-stage redirection chains can further bypass detection, since intermediate malicious content is not observable from the final URL representation. The model is also constrained by a fixed vocabulary size of 10,000 tokens, which introduces another limitation. This allows to exploit unseen or randomly generated subdomain patterns that are not properly represented during training. As a result, such inputs may be misclassified due to incomplete or weak token embeddings. The simplified attention mechanism, while computationally efficient, may not fully capture subtle adversarial perturbations embedded in URL structures. Consequently, carefully crafted character-level modifications can disrupt attention weights and reduce classification reliability.

The model has a problem with unseen data and if the attacks are totally new. New phishing tricks often use brand ways to make fake websites combine social engineering with other tricks or use computer generated URLs that are different from what we know. The model takes time to learn so it takes a while to figure out attacks. This means the system is not safe from attacks for some time. Also, people who make phishing websites try to make them look like websites, which makes it even harder for the model to find them. This distributional overlap makes zero-day detection even more challenging for feature-based learning.

Ethical Considerations and Risks

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05, May 2026
Received: 04-04-2026 Accepted: 09-05-2026 Published: 18-05-2026

The deployment of automated deep learning models for security filtering carries intrinsic ethical responsibilities and potential risks. The problem with false positives, if a deep learning model wrongly identifies a website as a phishing threat that can cause a lot of trouble. The people who own that website might get in trouble for no reason. That can hurt their reputation and cost them money. Second, the open-source and transparent nature of this architecture introduce a dual-use risk. Malicious actors could leverage knowledge of the model's structure to train Generative Adversarial Networks (GANs) that systematically craft adversarial URLs designed to exploit the model's blind spots, thereby escalating the cyber arms race.

Future Work

To address the limitations that are mentioned above, future work should focus on developing a more comprehensive and adaptive detection framework. A good way is to integrate **multi-modal inputs**, combining URL features with webpage content, HTML structure, DNS records, and TLS certificate information. This would enable the model to detect phishing attempts that are invisible at the URL level alone. Another improvement involves replacing the fixed vocabulary tokenizer with **subword or character-level encoding techniques**, which can significantly reduce out-of-vocabulary issues. This would improve the model's ability to handle unseen, rare, or newly generated URL patterns. To handle zero-day attacks more effectively, incorporating **continual or online learning strategies** would allow the model to adapt to newly emerging threats over time. This would reduce the delay between attack emergence and model retraining cycles. Additionally, combining supervised learning with **anomaly detection methods** could help identify previously unseen phishing patterns. In this area using a one-class learning mechanism can make it stronger by modeling only good URL behavior. Finally, exploring quantum-enhanced machine learning primitives offers a promising avenue for sequence analysis; integrating Variational Quantum Circuits (VQCs) or utilizing entirely quantum transformer models could theoretically accelerate the processing of vast, complex feature spaces beyond classical capabilities [9][10].

6. CONCLUSION

In this paper, we introduced a simplified, Transformer-based deep learning model utilizing the TensorFlow Keras Functional API for the critical task of phishing website detection. By utilizing an embedded Multi-Head Attention mechanism focused exclusively on URL character and token sequences, the proposed model can predict phishing websites well without using too much computer power. The empirical results from the initial training epochs demonstrate that even a heavily constrained self-attention network can achieve exceptional validation accuracy, outperforming traditional sequential models without the latency introduced by massive pre-trained language architectures.

The findings underscore the immense potential of tailored Transformer models in high-velocity cybersecurity environments. Cyber threats are getting more complicated all the time so we really need to have detection systems that are quick and can handle a lot of things accurately. Ultimately, this research provides a robust foundational framework for deploying lightweight attention networks in real-time threat detection systems, paving the way for future integrations with multi-modal security platforms and next-generation computational paradigms.

CONFLICT OF INTEREST

The Authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Li, Wenhao, Manickam, Selvakumar, Chong, Yung-Wey, Karuppayah, Shankar, Nanda, Priyadarsi, Li, Binyong, "PhishSSL: Self-Supervised Contrastive Learning for Phishing Website Detection," 2025. https://arxiv.org/pdf/2510.05900v1

[2] Gibadullin, Ilshat, Valeev, Aidar, "Experiments with LVT and FRE for Transformer model," 2020. https://arxiv.org/pdf/2004.12495v1

[3] Esfandiari, Nura, Kiani, Kourosh, Rastgoo, Razieh, "A Conditional Generative Chatbot using Transformer Model," 2023. https://arxiv.org/pdf/2306.02074v2

[4] Li, Wenhao, Manickam, Selvakumar, Chong, Yung-wey, Karuppayah, Shankar, "PhishDebate: An LLM-Based Multi-Agent Framework for Phishing Website Detection," 2025 IEEE International Conference on Big Data (BigData), Macau, China, 2025, pp. 6606-6615, 2025. doi:10.1109/BigData66926.2025.11401440

[5] Yu, Chang, Xu, Yongshun, Cao, Jin, Zhang, Ye, Jin, Yinxin, Zhu, Mengran, "Credit Card Fraud Detection Using Advanced Transformer Model," 2024. https://arxiv.org/pdf/2406.03733v4

[6] Ghosh, Subhadip, Gebru, Endalk Y., Kashyap, Chandramouli V., Harjani, Ramesh, Sapatnekar, Sachin S., "Accelerating OTA Circuit Design: Transistor Sizing Based on a Transformer Model and Precomputed Lookup Tables," 2025. https://arxiv.org/pdf/2502.03605v1

[7] Xiao, Jianli, Long, Baichao, "A Multi-Channel Spatial-Temporal Transformer Model for Traffic Flow Forecasting," Xiao J, Long B. A Multi-Channel Spatial-Temporal Transformer Model for Traffic Flow Forecasting[J]. Information Sciences, 2024: 120648, 2024. doi:10.1016/j.ins.2024.120648

[8] Reiser, Patrick, Eberhard, Andre, Friederich, Pascal, "Implementing graph neural networks with TensorFlow-Keras," Softw. Impacts 2021, 9, 100095, 2021. doi:10.1016/j.simpa.2021.100095

[9] Roosan, Don, Khan, Rubayat, Ashakin, Md Rahatul, Khou, Tiffany, Nirzhor, Saif, Haider, Mohammad Rifat, "Quantum Variational Transformer Model for Enhanced Cancer Classification," 2025. https://arxiv.org/pdf/2506.21641v1

[10] Khatri, Nikhil, Matos, Gabriel, Coopmans, Luuk, Clark, Stephen, "Quixer: A Quantum Transformer Model," 2024. https://arxiv.org/pdf/2406.04305v1

[11] A. Vaswani et al., "Attention Is All You Need," NeurIPS, 2017. https://arxiv.org/abs/1706.03762

[12] H. Le et al., "URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection," KDD, 2018. https://doi.org/10.1145/3219819.3219824

[13] O. K. Sahingoz et al., "Machine Learning Based Phishing Detection from URLs," Expert Systems with Applications, 2019. https://doi.org/10.1016/j.eswa.2018.09.029

[14] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," Google Research, 2016. DOI: https://doi.org/10.48550/arXiv.1605.08695

[15] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
DOI:
https://doi.org/10.7551/mitpress/10253.001.0001

[16] R. S. Rao and A. R. Pais, "Detection of Phishing Websites Using Automated Feature Extraction," International Journal of Information Management, 2019.
https://doi.org/10.1016/j.ijinfomgt.2018.08.005

[17] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers," NAACL, 2019. https://doi.org/10.18653/v1/N19-1423

[18] Prasad, A., & Chandra, S. (2023). PhiUSIIL Phishing URL Dataset [Dataset]. Kaggle. https://www.kaggle.com/datasets/ndarvind/phiusiil-phishing-url-dataset