2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC)

BaitNet: A Deep Learning Approach for Phishing Detection

1st Barath S
CSE with Cyber Security
Sathyabama Institute of Science and Technology
Chennai, India
barathsoccer15112@gmail.com

2nd Jaikrishnan J
CSE with Cyber Security
Sathyabama Institute of Science and Technology
Chennai, India
jaikris24@gmail.com

3rd Divas A S
CSE with Cyber Security
Sathyabama Institute of Science and Technology
Chennai, India
divagopi53@gmail.com

4th Harini S
CSE with Cyber Security
Sathyabama Institute of Science and Technology
Chennai, India
hariniswa2004@gmail.com

5th Nanthini N
CSE with Cyber Security
Sathyabama Institute of Science and Technology
Chennai, India
drnanthininsivarajan@gmail.com

2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC) | 979-8-3503-1702-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICMNWC60182.2023.10436016

***Abstract*—Recent years have witnessed a surge in phishing attacks, propelled by advancements in technology. This paper addresses the evolving threat landscape, particularly the rise in deception scams and highly sophisticated online attacks. This work presents a novel approach to detecting phishing sites using deep learning. Here, A URL-centric solution called BaitNet, employs a Convolutional Neural Network (CNN) model to significantly enhance accuracy in phishing website detection. BaitNet achieves outstanding performance metrics, including a precision rate of 98.99%, an F1 score of 99.00, and an accuracy of 98.44%. These results underscore its exceptional efficiency in safeguarding Internet users from phishing threats. This research underscores both technological innovation and the urgency of protecting individuals and organizations in an era of evolving cyber threats. As online threats grow more sophisticated, proactive defenses like BaitNet play a crucial role in upholding security and integrity of online activities, making it a pivotal advancement in cybersecurity.**

Keywords—Phishing, online attacks, Deep neural network, Convolutional neural network, URL.

I. Introduction

Phishing is a type of manipulation attack in which an attacker poses as a legitimate business and tricks targets into providing confidential information. Depending on the attacker, they can have a wide range of goals. In the entries, the term "phishing" was used by Khan, registered on January 2, 1996. The mention was in a Usenet newsgroup called AOHell [1]. This attack stands out among the most notorious cybercrimes, considering the staggering amount of around 3.4 billion spam messages in a single day. Most phishing attacks start with spam. These emails often contain links to phishing sites. The message appears to be from a trusted sender. When victims are tricked they are tricked into providing confidential information, usually on a phishing website. Sometimes malware is also downloaded to the target system [2]. In 2021, approximately 323,972 Internet users fell victim to phishing attacks.

Phishing attacks pose urgent and complex challenges in contemporary society, with many organizations lacking adequate anti-phishing methods to safeguard users against malicious Uniform Resource Locator (URLs). Leveraging deep learning techniques, particularly character-level Convolutional Neural Networks (CNNs), has emerged as an efficient means to identify malicious URLs rapidly [3]. The proposed model, BaitNet employs a 1D CNN model, encompassing embedding, Conv1D, max-pooling1D, flatten, and dense layers for URL classification. The model, evaluated on split datasets for training, validation, and testing, addresses the critical need for innovative cybersecurity approaches in the face of evolving cyber threats.

The main contributions of this paper are as follows:

- This research contributes significantly to the field of cybersecurity, addressing the need for innovative approaches against the expanding landscape of evolving and common cyberthreats.
- This work adopts an assertive stance to address unknown challenges, ensuring the continued effectiveness of the methods against emerging threats.
- This work enhances the user experience by enabling individuals to evaluate the legitimacy of websites themselves, empowering them to make informed decisions about online content.
- The project places a strong focus on establishing security precautions, with the proposed BaitNet serving as a vital component in safeguarding against phishing attacks.
- Ultimately, this research strives to protect the security and integrity of online activities in the face of emerging cyberthreats, contributing to a safer digital environment for all.

BaitNet uses innovative techniques in this project, including character-level tokenization for sophisticated URL representation, LeakyReLU activation for training robustness, and deep learning for pattern detection. Phishing detection is improved via an inventive exceeding technique. These developments, along with a remarkable accuracy rate, establish BaitNet as a potent and successful cybersecurity solution.

II. Literature Review

Many researchers have studied the results of phishing websites. This approach uses critical concepts from previous findings. The studied previous initiatives that used URL attributes to detect phishing, which influenced the current methodology. Jianguo Jiang et al [4] proposed a web-based malicious URL detection system by combining deep neural networks with natural language processing and threat intelligence. Asadullah Safi et al [5] did a Systematic

979-8-3503-1702-2/23/$31.00 ©2023 IEEE

Literature Review (SLR) survey on different phishing detection techniques and analyzed the performance of all those techniques and approaches. A novel method developed by Yao et al [6] that focuses on URL analysis has been identified as a precise and effective detection strategy for phishing websites.

Erzhou Zhu et al [7] proposed Online Feature Selection and Neural Network (OFS-NN), an effective phishing website detection model based on the optimal feature selection method and neural network. In the proposed OFS-NN, a new index Feature Validity Value (FVV) is presented for the first time to evaluate the effect of sensitive features on the detection of phishing sites. An algorithm is then created based on the new FVV index to select optimal features from phishing sites. Ammar Odeh et al [8] proposed a Neural Network with a multilayer perceptron to detect the scam URL. The proposed system improved the accuracy of the scam detection system as it achieved a high accuracy percentage of 98.5%.

Ningxia Zhang et al [9] proposed a detection model that incorporates a feed-forward neural network. Smita Sindhu et al [10] analyzed various machine learning methods that are used to detect phishing websites. Deep Neural Networks (DNN), Convolutional Neural Networks, and Recurrent Neural Networks (RNN)/Long Short-Term Memory (LSTM) Networks were the most frequently used deep learning algorithms, according to a systematic literature review conducted by Cagatay Catal et al [11]. The most effective deep learning-based algorithms were DNN and hybrid deep learning algorithms. Selamat et al [12] used logistic regression with CNN and CNN-LSTM to assess two URL data sets for phishing detection. They acquired information from a variety of sources, including phishing domains from Open Phish and malware domain listings from PhishTank.

III. Methodology

This code implementation develops a deep learning model to detect phishing URLs using a dataset of URLs labeled as "Legitimate" or "Phishing". The process includes character-level labeling, data processing, oversampling to correct class imbalance, and CNN model building at different levels. The model is trained, evaluated, and tested with performance metrics such as loss, accuracy, confusion matrix, and a classification report that evaluates its effectiveness in distinguishing between "legitimate" and "phishing" URLs. This computational approach focuses on reducing the risks associated with phishing attacks in online environments.

A. Datasets

To achieve a constructive phishing prediction model, training and testing data must consider target quality, credible source, and class richness, which greatly affect detection. Fig. 1, A dataset of legitimate and phishing URLs is gathered from various sources, which included 77,463 phishing websites and 38,232 legitimate real URLs. For both legal and phishing URLs, the dataset had a distinct source. PhishTank provided the source for the URLs of phishing websites, and took into account those that have been confirmed on the site as phishing. The primary concern with most of the reviewed papers is the limited dataset sizes. Small datasets can lead to evaluation challenges and sampling issues, reducing confidence in model assessment. Typically, the dataset was split into 80% for training and 20% for testing, and legitimate website URLs were sourced from platforms like Kaggle.

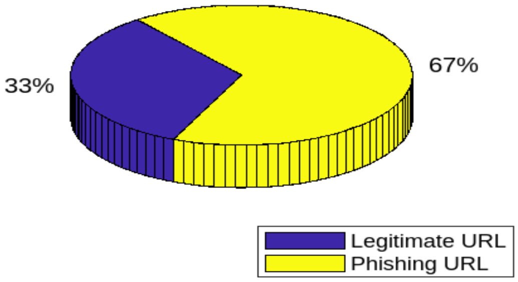


Fig. 1. Composition of Dataset

B. Data Preprocessing

To prepare the dataset for training the model several data preprocessing measures were taken,

1) Tokenization: Each URL is split into individual tags, where each character is treated as a separate character. This tokenization process initializes the Tokenizer with the parameter, limiting the vocabulary to the most common characters [13]. Here, the maximum vocabulary is set to maximum characters of 20,000 which ensures that URLs are represented as numerical sequences, making them suitable for learning tasks.

2) Sequencing: In this model, Sequencing involves matching the token against URL patterns and assigning each token a unique integer based on its frequency in the dataset [14]. This vocabulary is then used to convert tokenized URLs into numeric strings, replacing each character with a corresponding integer. This process prepares data for model training by representing URLs as sequences of integers.

3) *Padding:* Padding is applied to the numerical sequences of URLs in this project to ensure they all have a uniform length of 128 characters before being used as input for the deep learning model [15]. The process involves adding zeros to shorter sequences or truncating longer ones to a specified maximum length. Padding ensures consistent data dimensions for model training, allowing it to process the URLs efficiently regardless of their original lengths.

C. BaitNet Architecture

The model architecture begins with a character embedding approach, where characters within input data, presumably URLs, are transformed into numerical representations. These character embedding serves as the initial input for a sophisticated Convolutional Neural Network. Within this architecture, various layers and operations are employed to process the altered URL matrix, which probably represents a modified form of the input URLs. Fig. 2 provides a visual representation of the model's block diagram.

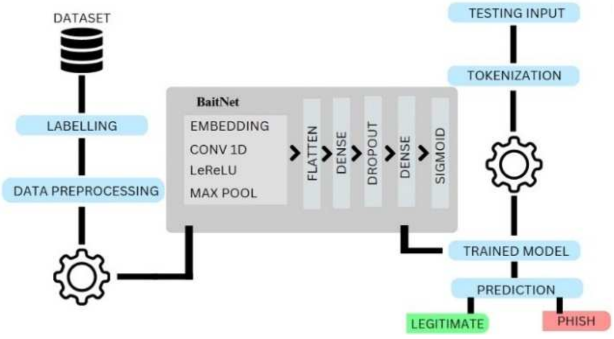


Fig. 2. Architecture of the Proposed work

1) Embedding Layer: The embedding layer plays a pivotal role in the deep learning architecture, as it transforms the input URL sequences into high-dimensional vectors. Specifically, the limit of the input length is 128 characters [16], which is often sufficient for capturing the essential features of URLs. In this layer, characters are mapped to vector representations, enabling the model to learn intricate patterns and relationships between characters within the URL data.

2) Convolution Layer: Conv1D layers are integral components responsible for extracting patterns and features from URL character sequences [17]. These layers employ convolution operations to scan the input sequences, using 64 filters, each with a kernel size of 3, to detect relevant patterns. The resulting feature maps represent the presence or absence of specific features within the URLs [18]. The convolution operation, represented by the formula,

$$Y[i] = \sum_k X[i+k]w[k] + B \qquad (1)$$

where,

Y[i] is the value at position *i* in the output feature map.

X[i] represents the input values at position *i* in the input feature map.

W[k] is the weight of the filter at position *k.*

B is the bias term.

Subsequently, MaxPooling layers are applied to reduce spatial dimensions while preserving significant information. This combination of convolution, activation, and pooling enables the model to learn hierarchical representations of URL data, enhancing its ability to distinguish legitimate from phishing URLs.

3) Activation Function: The activation functions within neural networks serves a vital role, introducing non-linearity to enable the network to capture intricate patterns in data. Activation functions are applied to each neuron's output in hidden layers [19]. To enhance training stability, Batch Normalization is often paired with activation function, while Dropout, although not technically an activation function, helps combat overfitting by randomly deactivating neurons during training, thus bolstering the model's resilience.

a) Sigmoid: The sigmoid activation function, often referred to as the logistic function, maps the model's final output to a value between 0 and 1. It has the following formula:

$$f(x) = \frac{1}{1+e^{-x}} \qquad (2)$$

where, f(x) represents the output of the sigmoid function and

x is the input to the function.

The primary objective of the sigmoid activation function is to classify the legitimate and phishing URLs using simplified binary classification method. And sigmoid activation function simplifies binary classification in phishing URL detection by mapping neural network output to a probability score between 0 and 1. A threshold, typically 0.5, categorizes URLs as either phishing (1) or legitimate (0).

b) LEAKYReLU (Leaky Rectified Linear Unit): In this CNN model, LeakyReLU is applied just after the first convolutional layer This activation function is characterized by the formula,

$$f(x) = \{x, if\colon x \geq 0 \; \alpha x, if\colon x < 0 \qquad (3)$$

where, f(x) signifies the output for an input value x.

When x is greater than or equal to zero, LeakyReLU permits it to pass through unaltered, retaining its value. Conversely, when x is negative, the function allows a small fraction (α) of the negativity to "leak" through. It deals with the "dying ReLU" problem, where neurons may become inactive (output zero for all inputs) during training, which is a potential issue with the regular ReLU activation [20]. This subtle nonlinearity introduced by LeakyReLU is instrumental in mitigating the vanishing gradient issue during training. By incorporating LeakyReLU in this work, it significantly bolsters the model's capacity to discern subtle patterns in URL character sequences, elevating the accuracy and efficacy of URL phishing detection.

4) Max Pooling: MaxPooling is a down sampling technique that helps capture and retain the most relevant information from the convolutional layers while reducing the dimensionality of the data. MaxPooling 1D operates along the temporal dimension of the data (URL sequences) and involves sliding a fixed-size window over the sequence. At each position of the window, it selects the maximum value, effectively summarizing the most salient features within that window. The key formula used in MaxPooling 1D is:

$$MaxPooling(x)[i] = max(x[i\colon i + poolsize]) \qquad (4)$$

Where, x represents the input sequence and i denotes the current position of the sliding window.

By applying this operation, MaxPooling 1D retains the most significant information while reducing the dimensionality of the feature maps. This aids in focusing on the most important features for subsequent layers, contributing to the model's ability to identify phishing URLs effectively.

5) Flatten Layer: The Flatten layer is used to convert the multidimensional output of the previous layer to a one-dimensional array. It is necessary before sending the output to a dense layer that is fully linked and only accepts one-dimensional inputs. The Flatten layer is typically employed in neural network architectures when transitioning from convolutional or recurrent layers to fully connected layers [21]. The Flatten layer enables the smooth transition from the convolution layer's output to these dense layers input, allowing the model to leverage the features extracted by the convolutional layer for making predictions.

6) Dense Layer: The dense layer has a complex connection to the layer above it, emphasizing the fact that every neuron in the layer is connected to every neuron in the layer above it. Fully connected layers are crucial for combining the information acquired from previous layers and generating conclusive predictions [22]. In Baitnet, the first dense layer using LeakyReLU as the activation function consists of 128 neurons. Each of these neurons is working to recognize different aspects of web interaction while also collecting complicated patterns and interactions. The validation of positive output for advantageous patterns by this function makes the model more intuitive. The second dense layer of sigmoid activation renders an absolute conclusion [23]. It generates a single output that, using just one unit, indicates the model's confidence that the web link is either real (near 0) or phishing (close to 1). The key formula used in Dense Layer is:

$$Y = W * X + B \tag{5}$$

Where, Y signifies the output, W represents the weight matrix, X is the input, and B is the bias term. By utilizing Dense layers, the model effectively captures complex dependencies within the data, ultimately enhancing its capacity to discriminate between legitimate and phishing URLs.

7) Dropout Layer: The Dropout layer strengthens the model and enhances the model's capacity to adapt a new untested URLs, limiting the effect of individual neurons during training and preventing overfitting [24]. During each training batch, this Dropout layer randomly removes 50% (0.5) of the neurons. By preventing the neural network from fitting noise in the training data, it operates as a type of regularization and enhances its ability to generalize to new data [25,28]. The performance of the model is influenced by additional factors like activation function, kernel size, and batch normalization. Table I shows the paramaters generated during the training of the proposed model.

TABLE I. PARAMETER TABLE OF THE PROPOSED BAITNET MODEL

LAYER	INPUT SHAPE	OUTPUT SHAPE
Embedding	(None, 128)	(None, 128, 300)
Conv1D	(None, 128, 300)	(None, 126, 64)
LeakyRelu	(None, 126, 64)	(None, 126, 64)
Batch Normalization	(None, 126, 64)	(None, 126, 64)
MaxPooling	(None, 126, 64)	(None, 63, 64)
Flatten	(None, 63, 64)	(None, 4032)
Dense	(None, 4032)	(None, 128)
LeakyRelu	(None, 128)	(None, 128)
Batch Normalization	(None, 128)	(None, 128)
Dropout	(None,128)	(None, 128)
Dense	(None,128)	(None, 1)

Total Parameters: 6,574,785

Trainable Parameters: 6,574,785

Non-Trainable Parameters: 0

IV. RESULT AND DISCUSSION

A. System and Software Requirements

This proposed method was executed on a well-equipped system, featuring high-performance hardware, including an NVIDIA RTX 3080 Ti graphics card with 8GB of memory, an Intel Core i7 11th Gen processor, and 32GB of RAM. The ASUS B406 motherboard was employed to support these hardware components. The programming language used was Python, specifically version 3.11.3, complemented by libraries like TensorFlow and scikit-learn (sklearn) for machine learning and data analysis. MATLAB was utilized for data visualization. Furthermore, the utilized Flask framework is established a seamless connection to the backend of the system, enabling efficient data processing. To provide a user-friendly interface for interaction, the integrated HTML and JavaScript, facilitating real-time interaction and usability. This overview highlights the technical resources utilized in this research.

B. Training Phase

In this section, the training phase of the proposed BaitNet model is discussed. This model employs a 1D CNN architecture with a specific layer configuration, including Embedding, 1D Convolution, Activation function, MaxPooling, Flatten, Dense and Dropout. The comprehensive analysis of the training process includes the examination of key performance metrics, such as training and validation accuracy and loss. Additionally, the Receiver Operating Characteristic (ROC) graph and the confusion matrix provides a holistic view of the model's performance. These visualizations help in assessing the model's effectiveness. Notably, the proposed BaitNet model achieves an impressive accuracy of 98.44%, as demonstrated by the graph and metrics.

From Fig. 3, it is shown that the model has a high training accuracy and validation accuracy which indicates that the model works very well. The difference between the predicted results of the classification model and the actual target values for a given training dataset is measured as the training loss which is shown in Fig. 4. It is used as an optimization goal during training to minimize training losses so that the model forecasts are as close as possible to actual targets. Less validation loss means that the model better matches the learning algorithm. On the other hand, Low training loss indicates that the model has effectively minimized the difference between its predicted outputs and the true target values in the training dataset.

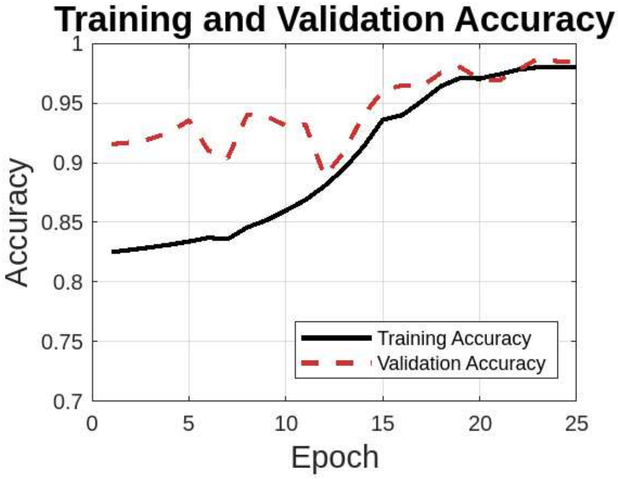


Fig. 3. Training and Validation Accuracy of the Proposed BaitNet Model

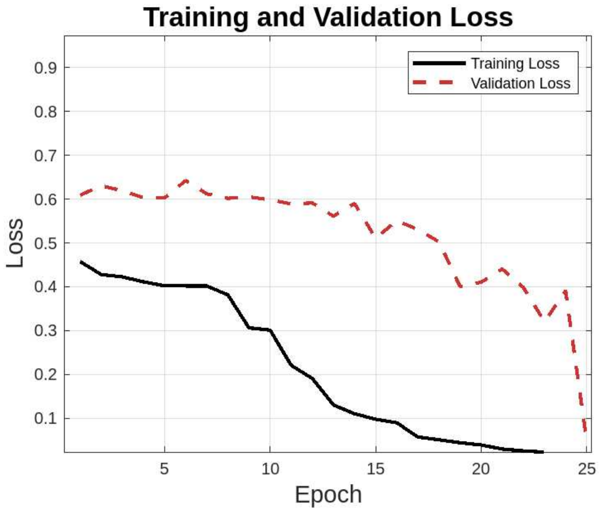


Fig. 4. Training and Validation Loss of Proposed BaitNet Model

Fig. 5 shows the confusion matrix for BaitNet Model, which assesses the classification model's accuracy by quantifying False Positive (FP), True Positive (TP), False Negative (FN), and True Negative (TN) predictions. TN represents correct identification of negative samples, while FN indicates positive samples mistakenly labeled as negative. TP signifies accurate identification of positive samples, and FP measures incorrect categorization of negative samples as positive. All four metrics are crucial for evaluating the model's effectiveness.

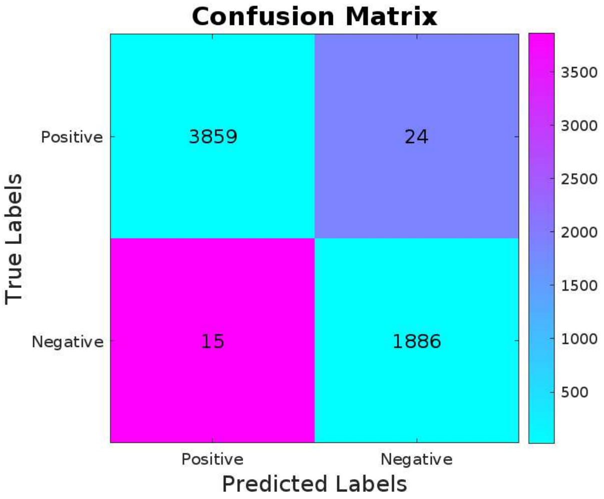


Fig. 5. Confusion Matrix of proposed BaitNet Model

The Receiver Operating Characteristic (ROC) curve depicts a binary classifier's performance across various discrimination thresholds. It compares True Positive Rate (TPR) with False Positive Rate (FPR) as the threshold changes. A perfect classifier yields a TPR of 1 and FPR of 0, creating a point in the upper left corner of the ROC space. A random classifier forms a diagonal line from (0,0) to (1,1). BaitNet's ROC curve, with an impressive Area Under the Curve (AUC) of 0.98, signifies strong performance in distinguishing between legitimate and phishing URLs. A high AUC near 1 showcases the model's accuracy, highlighting its potential in real-world cybersecurity, especially in identifying and mitigating phishing threats. Fig. 6 provides a visual representation of the ROC curve.

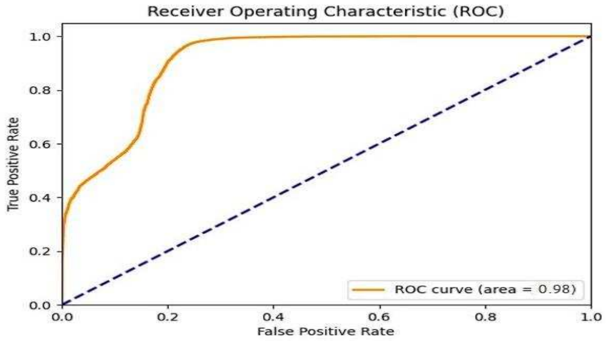


Fig. 6. ROC Curve of proposed BaitNet model

C. Performance Evalutaion

In this section, the BaitNet model is thoroughly examined and analyzed the performance and comparison to existing phishing detection models. Scrutinizing key metrics such as precision, accuracy, F1 score, sensitivity, specificity, and the Matthews Correlation Coefficient (MCC) helps uncover BaitNet's comparative strengths and advantages in the realm of phishing detection.

The proposed BaitNet model is compared to the state-of-the-art techniques, including RF Classifier, LSTM-CNN, XGBoost and SVM. RF classifier: By selecting important features from a collection of a period that improved the accuracy, precision, and recall of the Random Forest classifier. This emphasizes how crucial it is to determine the most important characteristics in order to distinguish phishing URLs. The Random Forest Classifier technique used in the work lead by Z. Zhang et al., produced multiple excellent metrics. LSTM-CNN: Phishing website detection achieved remarkable outcomes, according to the examination of the suggested system. The performance of the proposed deep learning methods differed. LSTM–CNN and LSTM excelled in both. XGBoost: This model presented a machine-learning method that assesses text, image, and URL data to detect ongoing phishing attacks. With the help of XGBoost, the results indicated assurance. This demonstrates the way it works against current phishing threats. SVM: Support vector machine (SVM)-supervised machine learning based methodology used by Dogukan Aksu et al., produced excellent performance against phishing threats. Table II and Table III shows the Performance comparison of the proposed BaitNet model with existing models. The Accuracy (ACC) of a model is calculated as the ratio of the number of correctly predicted instances (both true positives and true negatives)

$$Accuracy(\%) = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \qquad (6)$$

The proposed BaitNet model has an accuracy of 98.44%, which signifies an overall effective classification of both positive and negative instances. This ensures robust performance in differentiating between phishing and legitimate websites. In contrast, LSTM-CNN achieves an impressive accuracy of 97.60%, the small difference implies a potential margin for misclassification, which may impact the model's overall reliability. Precision, representing the proportion of true positive predictions among all positive predictions, is an important measure of a model's ability to make accurate positive identifications. Precision (PREC) is calculated using the formula:

$$Precision(\%) = \frac{TP}{TP+FP} \times 100 \qquad (7)$$

For the proposed BaitNet model, the exceptionally high precision (98.99%) implies a minimal rate of false positives. This is crucial in phishing detection as it ensures that when the model identifies a website as malicious, it is highly likely to be accurate. In contrast to RF Classifier, while achieving commendable precision at 92.13%, may have a slightly higher chance of false positives. This could lead to misclassifying benign websites as malicious. Another key metric is the F1 score, which harmonizes precision and recall. The F1 score considers both false positives and false negatives and is particularly useful when dealing with imbalanced datasets. The formula for F1 score is:

$$F1\,Score = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \qquad (8)$$

The proposed BaitNet model has an outstanding F1 score of 99.00% indicates a balanced trade-off between precision and recall. This balance ensures that the model not only identifies positives accurately but also captures a high percentage of actual positives. LSTM-CNN, with a slightly lower F1 score of 96.70%, may face challenges in maintaining an optimal balance between precision and recall, potentially resulting in either missed phishing websites or false positives.

TABLE II. PERFORMANCE COMPARISON OF PROPOSED BAITNET MODEL WITH EXISTING MODELS

AUTHOR	APPROACH	PREC (%)	RECALL (%)	F1(%)	ACC (%)
Z. Zhang et al., [26]	RF Classifier	92.13	91.54	91.83	b 92.90
R. Alaqel et al., [27]	LSTM-CNN	96.90	98.20	96.70	97.60
Rashid Amin et Al., [27]	XGBoost	90.59	93.00	91.66	91.56
Dogukan Aksu et al., [29]	SVM	91.66	88.00	89.79	95.00
Proposed Model	BaitNet	98.99	98.98	99.00	98.44

The proposed BaitNet model has sensitivity of 98.21% indicates a high true positive rate, ensuring the model captures a significant portion of actual phishing websites. RF Classifier, with a sensitivity of 89.43%, might miss a proportion of phishing websites, leading to false negatives and compromising the model's effectiveness in identifying malicious content. Due to its great sensitivity, BaitNet minimizes false negatives by capturing a significant percentage of true phishing sites. The true positive rate, or sensitivity, is a metric used to assess a model's accuracy in identifying positive cases. The formula for Sensitivity is:

$$Sensitivity = \frac{TP}{(TP+FN)} \qquad (9)$$

Specificity is another performance metric in binary classification, and it measures the model's ability to correctly identify negative instances. Its high specificity highlights how well it classifies authentic websites. The formula for Specificity is:

$$Specificity = \frac{TN}{(TN+FP)} \qquad (10)$$

The specificity of the proposed model is 99.38%, which emphasizes the ability to avoid false positives, contributing to a more secure identification of benign websites while LSTM-CNN maintains specificity of 98.26%. The Matthews Correlation Coefficient (MCC) is a balanced metric for evaluating binary classification models. It takes into account both true positive and true negative predictions, making it suitable for imbalanced datasets. The formula for MCC is:

$$MCC = \frac{(TP \times TN)-(FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (11)$$

The proposed model has a high MCC of 98.47%, which reflects strong overall prediction performance, considering both false positives and false negatives. RF Classifier, with an MCC of 84.66%, suggests a good but comparatively lower level of overall prediction robustness. This may result in a less reliable model.

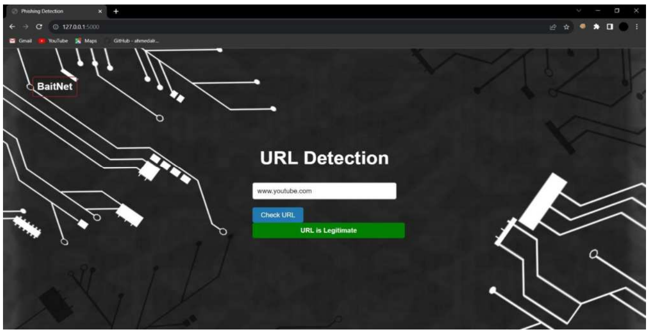
TABLE III. PERFORMANCE COMPARISON OF PROPOSED BAITNET MODEL WITH EXISTING MODELS

AUTHOR	APPROACH	SENSITIVITY	SPECIFICITY	MCC
Z. Zhang et al.,	RF Classifier	89.43	90.38	84.66
R. Alaqel et al.,	LSTM-CNN	96.98	98.26	95.26
Rashid Amin et al.,	XGBoost	92.57	90.59	83.14
Dogukan Aksu et al.,	SVM	91.67	96.05	86.52
Proposed Model	BaitNet	98.21	99.38	98.47

D. User Interface

The proposed BaitNet model is thoroughly examined in terms of user experience and system functionality. The system satisfies the requirements of individuals, interacts with them and transfers knowledge. The results of processing a valid URL using the BaitNet model, which returns a 'URL is Legitimate' classification, is shown in Fig. 7. This builds assurance in the model's practical use by showcasing its capacity to precisely identify and validate safe web addresses. Alternatively, Fig. 8 shows the way the model works to identify malicious URLs because it labels the URL as "Phishing" immediately, providing a protective guard against malicious and fraudulent websites. These illustrations not only validate the competence of the model and also highlight the accuracy of the model.

With to its outstanding performance metrics, BaitNet distinguishes out from other techniques for providing a dependable and secure solution to detect phishing websites. Because of its 1D CNN architecture, which is competent at identifying complex patterns in data, it is a powerful tool against internet scams and unethical activity. BaitNet is the best-performing approach in this comparison evaluation. Small metric adjustments show just how important it is to select a model that fits expected detection requirements.

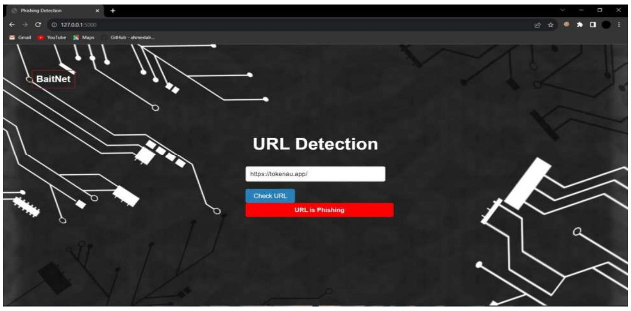


URL Detection

www.youtube.com

URL is Legitimate

Fig. 7. Output Screenshot of proposed BaitNet Model detecting Legitimate URL



URL Detection

https://tokenau.app/

URL is Phishing

Fig. 8. Output Screenshot of proposed BaitNet Model detecting Phishing URL

V. Conclusion

In this research, a novel deep learning algorithm called BaitNet is developed for URL phishing detection. This involved meticulous data preprocessing, which included tokenization, sequencing, and padding. The preprocessed data was then fed into the proposed BaitNet model, consisting of one Convolutional 1D Layer, a MaxPooling Layer, three different Activation Layers, a Flatten Layer, and two fully connected Layers, to identify legitimate and phishing URLs. With an F1 score of 99.00, an accuracy of 98.44%, and a precision rate of 98.99%, BaitNet demonstrates remarkable performance. BaitNet not only achieves exceptional accuracy but also shows strength in correct positively identifications and a balance between precision and recall. It also performs exceptionally well in precision and F1 score. Testing, high training, and validation accuracy show how effective the model performs. These outcomes highlight BaitNet's effectiveness in online security by demonstrating its capacity as a reliable and accurate tool for phishing attack detection.

Although, the proposed BaitNet model excelling in the performances, it also faces limitations. Generalizing to new attack strategies and sensitivity to URL representation pose additional concerns. The oversampling technique and limited consideration of non-URL features may result in potential false positives or negatives. Future work involves implementing dynamic learning mechanisms for real-time adaptation, exploring advanced URL representation techniques, refining oversampling methods, and incorporating non-URL features to enhance detection capabilities.

References

[1] Hamzah Salah, Hiba Zuhair "Deep learning in phishing mitigation: a uniform resource locator-based predictive model", Oct 2022, inpress.

[2] Birendra Jha, Medha Atre, Ashwini Rao, "Detecting Cloud-Based Phishing Attacks by Combining Deep Learning Models", Oct 2022

[3] Gopinath, M.; Sethuraman, S.C. ‘A Comprehensive Survey on Deep Learning Based Malware Detection Techniques.’ , December 2022

[4] Jianguo JiangJiuming Chen, Kim-Kwang Raymond Choo Chao Liu1 Kunying Liu1 Min Yu and Yongjian Wang "A Deep Learning Based Online Malicious URL and DNS Detection Scheme" pp. 438–448, 2018.

[5] Asadullah Safi, Satwinder Singh "A systematic literature review on phishing website detection techniques", February 2023.

[6] Wenbin Yao, Yuanhao Ding, Xiaoyong Li "Deep Learning for Phishing Detection"March 2019.

[7] Erzhou Zhu, Yuyang Chen, Chengcheng Ye, Xuejun Li, Feng Liu "OFS-NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network" June 2019.

[8] Ammar Jamil Odeh, Ismail Keshta, Eman Abdelfattah "Efficient Detection of Phishing Websites Using Multilayer Perceptron", July 2020.

[9] Ningxia Zhang, Yongqing Yuan "Phishing Detection Using Neural Network" , unpublished

[10] Smita Sindhu, Sunil Parameshwar Patil, Arya Sreevalsan, Faiz Rahman, Ms. Saritha A. N "Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation", December 2020.

[11] Cagatay Catal, Görkem Giray, Bedir Tekinerdogan, Sandeep Kumar, Suyash Shukla "Applications of deep learning for phishing detection: a systematic literature review", June 2022.

[12] Nguyet Quang Do, Ali Selamat, Ondrej Krejcar Takeru Yokoi and Hamido Fujita "Phishing Webpage Classification via Deep Learning-Based Algorithms: An Empirical Study", October 2021.

[13] Ganesh Lokare, ‘Preparing Text Data for Transformers: Tokenization, Mapping and Padding’, February 2023.

[14] Jianguo Jiang, Jiuming Chen, Kim-Kwang Raymond Choo, Chao Liu, Kunying Liu, Min Yu & Yongjian Wang ‘A Deep Learning Based Online Malicious URL and DNS Detection Scheme’, April 2018.

[15] Ali Moslah Aljofey, Dr. Qingshan Jiang, Dr. Jean-Pierre Niyigena, Qiang Qu, Dr. Mingqing Huang "An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL", September 2020.

[16] N. Nanthini, N. Puviarasan, P. Aruna, "Eye Blink-Based Liveness Detection Using Odd Kernel Matrix in Convolutional Neural Networks", International Conference on Innovative Computing and Communications (pp.473-483), January 2022.

[17] N. Nanthini, N. Puviarasan, P. Aruna, "A novel Deep CNN based LDnet model with the combination of 2D and 3D CNN for Face Liveness Detection", 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), July 2022.

[18] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, Daniel J. Inman “1D convolutional neural networks and applications: A survey”Version of Record November 2020.

[19] Shraddha Goled, “How Do Activation Functions Introduce Non-Linearity In Neural Networks?”, November 2021.

[20] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie & Laith Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", March 2021.

[21] Dr. Eman Abdullah Aldakheel, Dr. Mohammed Zakariah, Ghada Abdalaziz Gashgari, Fahdah A. Almarshad, Dr. Abdullah Ibrahim Alzahrani “A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators”, April 2023.

[22] Yugesh Verma, “A Complete Understanding of Dense Layers in Neural Networks”, September 2022.

[23] Arden Dertat “Applied Deep Learning - Part 1: Artificial Neural Networks “, Aug 2017.

[24] Imrus Salehin "A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain", July 2023.

[25] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, Ruslan R. Salakhutdinov, " Improving neural networks by preventing co-adaptation of feature detectors",July 2012.

[26] Shinelle Hutchinson, Zhaohe Zhang and Qingzhong Liu "Detecting Phishing Websites with Random Forest" 2018.

[27] Zainab Alshingiti Rabeah Alaqel Jalal Al-Muhtadi, Qazi Emad Ul Haq, Kashif Saleem and Muhammad Hamza Faheem "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN", January 2023.

[28] Muhammad Waqas Shaukat, Rashid Amin, Muhana Magboul Ali Muslam , Asma Hassan Alshehri and Jiang Xie "A Hybrid Approach for Alluring Ads Phishing Attack Detection Using Machine Learning" ,September 2023.

[29] Dogukan Aksu, A. I. Abdulwakil, M. Aydin "Detecting Phishing Website Using Support Vector Machine Algorithm", June 2017.