iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** X **Month of publication:** October 2023

DOI: https://doi.org/10.22214/ijraset.2023.56143

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue X Oct 2023- Available at www.ijraset.com

Spam Email Detection Using Convolutional Neural Networks: An Empirical Study

Akshay Merugu[1], Hrishikesh Goud Chagapuram[2], Rahul Bollepalli[3]
[1, 3]*CMR Institute of Technology, Hyderabad,*
[2]*Sreenidhi Institute of Science & Technology, Hyderabad*

Abstract: This study leverages Convolutional Neural Networks (CNNs); a state-of-the-art deep learning architecture primarily used in image analysis, and adapts it for the detection of phishing emails. By treating email content as multi-dimensional data, we employ CNNs to extract meaningful features and patterns from email headers, text, and attachments. Our approach not only identifies known phishing templates but also has the capability to detect emerging and zero-day phishing attacks.

I. INTRODUCTION

Phishing attacks remain a pervasive and evolving threat in the digital landscape, exploiting human vulnerabilities to deceive individuals and organizations into divulging sensitive information. In response to this escalating cyber menace, this research focuses on the development of a novel approach termed "Phishing CNN" for the automated detection of fraudulent emails.

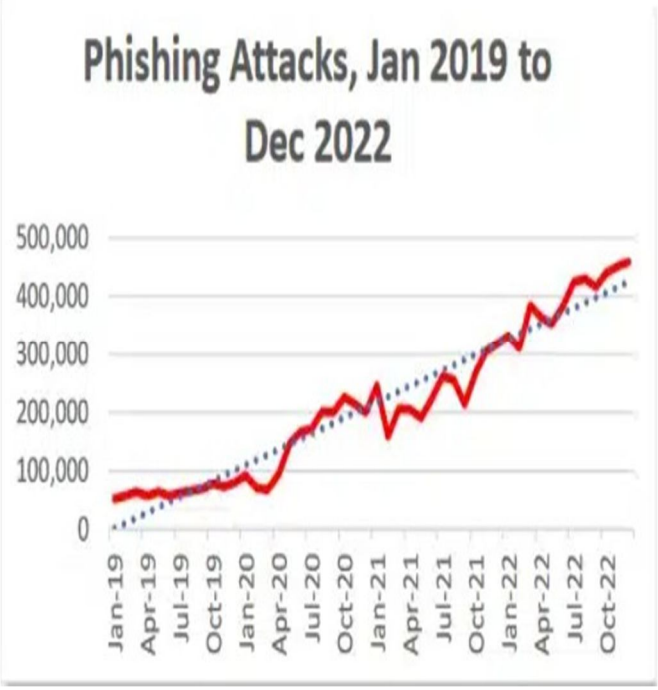


Fig. 1 Image showing the number of phishing attacks made through email in the past few years

To enhance the robustness and accuracy of our model, we explore various data preprocessing techniques, feature engineering strategies, and transfer learning from related tasks. Furthermore, we delve into the integration of natural language processing (NLP) techniques to analyze email text and identify subtle linguistic cues that may indicate phishing attempts.

The evaluation of Phishing CNN is carried out on a diverse and large-scale dataset, incorporating real-world phishing emails and legitimate correspondence. Our results demonstrate promising accuracy rates, low false positive rates, and excellent generalization performance, positioning Phishing CNN as a valuable tool in the fight against phishing attacks.

Ultimately, this research contributes to the arsenal of cybersecurity tools, offering a reliable and automated approach to detect fraudulent emails, thereby safeguarding individuals and organizations against the financial, reputational, and security risks associated with phishing threats.

II. LITERATURE REVIEW

Upon Extensive Literature Survey, the previous researches were based on how CNN algorithms and deep learning techniques are involved in detection of phishing and spam emails.

One such research is:

"A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN" by few students based in Saudi Arabia (detailed references are in later section of the paper)

The study conducted by these students aimed on using deep learning techniques namely CNN, LSTM (Long Short-Term Memory) which is a recurrent neural network (RNN) architecture widely used in Deep Learning. The study aimed to classify phishing URLs and stop financial losses and cybercrimes, our work offers a great contribution to the efficacy of using LSTM, CNN, and LSTM–CNN. Even though this paper is mostly based on solving phishing attacks through a particular source i.e.: emails, the paper mentioned above gives a meaningful insight on how these models usually work. This work aimed to classify phishing URLs and combat financial losses and cybercrimes.

Our project builds upon these foundations, further enhancing the efficacy of LSTM, CNN, and LSTM-CNN in the context of email security, thus contributing to the ongoing battle against phishing attacks.

Another paper from students of Sichuan University from China:

"Phishing Email Detection Using Improved RCNN Model with Multilevel Vectors and Attention Mechanism"

This is an advanced research on how RCNN Model with Multilevel Vectors and Attention Mechanism improves the currently deployed models used in several places of the internet. This model proposed a new phishing email detection model named THEMIS, which is used to model emails at the email header, the email body, the character level, and the word level simultaneously. To evaluate the effectiveness of THEMIS, we use an unbalanced dataset that has realistic ratios of phishing and legitimate emails which comprehensively models emails at various levels, including the email header, email body, character level, and word level. This innovative approach has been instrumental in improving existing models deployed across various facets of the internet. The evaluation of THEMIS against an unbalanced dataset with realistic ratios of phishing and legitimate emails demonstrates its potential to enhance email security significantly. These research contributions collectively underscore the continuous evolution and innovation in the domain of phishing email detection.

There are several other studies and researches conducted on this topic, whilst many aim to improve the model through several other techniques, the paper being produced here aims to drastically improve the accuracy and other metrics of detection using NLP (Natural Language Processing) and Sequence padding.

III. CNN METHODOLOGY

A. Neural Networks

Imagine a brain-like system that can learn from examples, make decisions, and solve complex problems. At its core, a neural network is a collection of interconnected nodes, often referred to as "neurons." These neurons work together to process information, just like our brain's neurons. Each neuron receives inputs, performs computations, and produces an output. When combined, these neurons can perform tasks ranging from recognizing images to playing games and making predictions. That's the essence of a neural network—a powerful computational tool inspired by the human brain.

Neural networks are typically organized into layers: an input layer, one or more hidden layers, and an output layer. Think of these layers as processing stages. The input layer receives data (like pixel values in an image), the hidden layers analyze and transform this data, and the output layer produces a final result (like classifying an image as a cat or a dog).

B. Layers of Neurons

Neural networks are typically organized into layers: an input layer, one or more hidden layers, and an output layer. Think of these layers as processing stages. The input layer receives data (like pixel values in an image), the hidden layers analyze and transform this data, and the output layer produces a final result (like classifying an image as a cat or a dog).

C. Connections and Weights

Connections between neurons are like synapses in our brains. Each connection has a "weight" that determines its strength. These weights are crucial because they influence how information flows through the network. During training, the network adjusts these weights to learn from data.

D. Activation Functions

Neurons use activation functions to decide whether to "fire" or pass information to the next layer. Common activation functions include the sigmoid, ReLU (Rectified Linear Unit), and tan h (Hyperbolic Tangent). These functions introduce non-linearity, allowing neural networks to learn complex patterns.

IV. COMPLICATIONS OF USING TRADITIONAL ANN'S FOR IMAGE CLASSIFICATION:

For a general image of small size, using a artificial neural network with multiple hidden layers for activation can be considered. But, as the size of the image (size here refers not only to the actual dimensions of the image but also to the definition of the image and the number of features involved in the image) Images are high-dimensional data, often with millions of pixels. Traditional ANNs may struggle to handle this high dimensionality, leading to a large number of weights and parameters, making training slower and more prone to overfitting. ANNs are prone to overfitting, especially when dealing with small or noisy datasets. Over fit models perform well on training data but poorly on new data because they capture noise and outliers. Processing large images with traditional ANNs can be computationally intensive and may require extensive down sampling or cropping to reduce dimensionality. This can result in information loss and hinder the network's ability to recognize fine details. The image requires much more computation power to adapt itself to the image and requires a dense layer of neural connections this in turn puts heavy strain on the CPU.

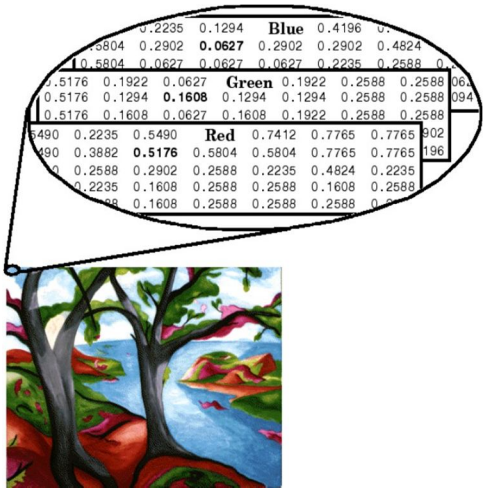
V. CNN OVER TRADITIONAL ANN

Convolutional Neural Network works in two stages namely:

A. Feature Extraction

Features are the particular trait of the image which the convolutional neural network looks for while matching with other images during the classification phase. These feature Extraction involves multiple phases of convolutional methods along a reduction phase to decrease the number of computations involved using a method called pooling.

Once an image is loaded into the network as any neural network it first converts itself into a matrix of RGB values which present the values of the colors being depicted in the picture.



As being displayed in the above image, the whole picture is initially converted into a matrix of numbers representing the RGB values of the color. Later during the process of convolution the image is then extracted of features which represent the picture for example in the given picture the branches, the water and many others. These are initially converted into matrices of values.

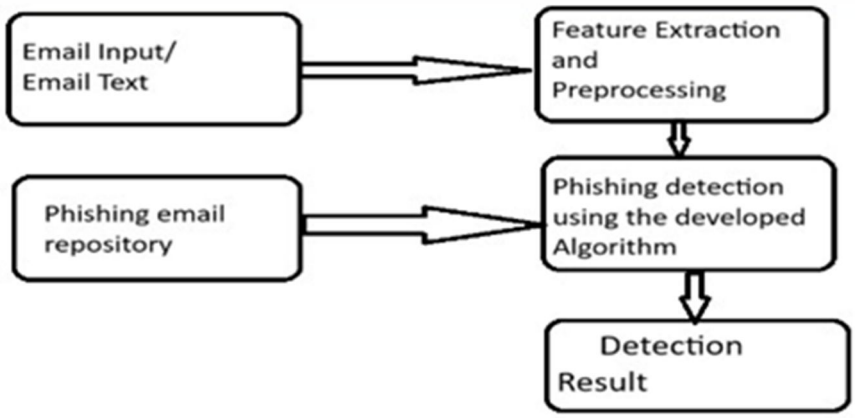
B. Convolution

In the convolution phase, the image which is being detected is being plotted (multiplied) with the cells of the matrix of a feature extracted from the initial image. This then checks the values which indicate whether the value is present or not indicating the presence of the feature.

This is then applied for all the features which are then again checked for major features involving the previous ones and then a neural network is formed to classify the given image into one category.
Pooling is the process of considering only the maximum values in a stride, usually a lot smaller than the feature matrix. Various levels of pooling and convolution take place to reduce the computations required to classify the image.

VI. EXISTING SYSTEM

The Modern systems are actively updating in terms of neural networks, yet the architecture behind it remains the same, A system used for phishing somewhat looks like this:



A. *Email Input*

The email text is provided as input to the system.

B. *Feature Extraction and Preprocessing*

This stage involves extracting relevant features from the email text, such as text content, sender information, attachments, etc. The data is preprocessed to make it suitable for analysis.

C. *Phishing Detection:*

This step uses existing phishing detection techniques, which could be based on machine learning models or rule-based systems. These systems analyze the extracted features and determine whether the email is phishing or not.

D. *Detection Result:*

The final output is whether the email is classified as phishing or not phishing. The existing email detection has many flaws, while the pro's cannot be overlooked as these kinds of systems are being applied or used over many industries and organizations to protect themselves from many kinds of phishing attacks. These systems typically aim for a standard template of phishing mails which try to detect them and stop them before being delivered to the end user. There have been several upgrades to these systems where they can now detect newer type of emails which are sent with malicious intent. Yet, features such as headers, text images, URL's, ASCII codes can still mislead the system into believing them that they are legit emails and not phishing. Several other limitations are explored in detail in the next section.

VII. LIMITATIONS OF EXISTING SYSTEM

A. *False Positives and False Negatives*

Existing systems may generate false positives (legitimate emails classified as phishing) and false negatives (phishing emails classified as legitimate). Achieving a balance between these two types of errors is challenging.

B. *Evolution of Phishing Techniques*

Phishers continually adapt and develop new techniques to evade detection. Existing systems may struggle to keep up with evolving phishing tactics.

C. Zero-Day Attacks

Rule-based systems may fail to detect zero-day phishing attacks that employ entirely new strategies, as these systems rely on predefined rules or patterns.

D. Imbalanced Datasets

Machine learning-based systems require large and balanced datasets for training. In practice, obtaining representative datasets with sufficient phishing examples can be challenging.

E. Feature Engineering

Traditional machine learning approaches often require manual feature engineering, which can be time-consuming and may not capture all relevant features.

F. Lack of Generalization

Some systems may perform well on specific types of phishing attacks but may struggle to generalize to different variations or new types of attacks.

G. Overfitting

Machine learning models can over fit to the training data, leading to poor performance on unseen data.

H. Resource Intensive

Some machine learning models, especially deep learning models, can be computationally expensive and may not be feasible for all organizations, especially smaller ones with limited resources.

I. Privacy Concerns

Some phishing detection systems may involve the analysis of email content, raising privacy concerns related to user data.

J. Scalability

As the volume of email traffic grows, scalability becomes a concern for some systems. Scalable deployment and real-time detection can be challenging.

K. Learning over Time

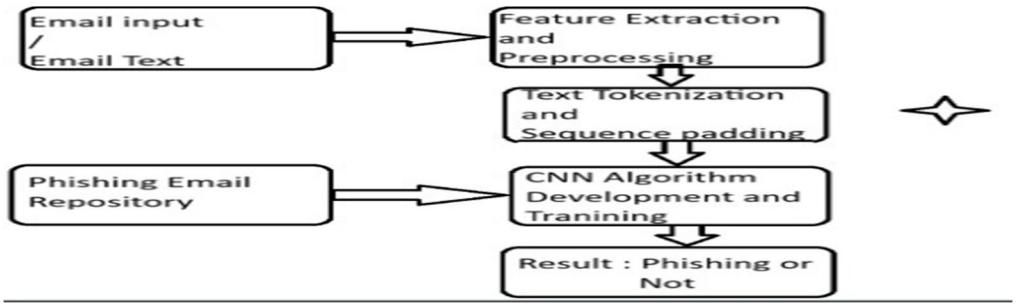
Machine learning models like CNNs can continuously improve their performance as they receive more data and feedback. This allows for ongoing refinement and adaptation to changing phishing techniques.

L. Potential for Real-Time Detection

Once trained, CNN-based models can make predictions in real-time, providing immediate detection of phishing emails.

VIII. PROPOSED SYSTEM

Building upon the foundation of existing phishing email detection techniques and leveraging Convolutional Neural Networks (CNNs), we propose an advanced system that enhances the accuracy and effectiveness of identifying phishing emails. Our system aims to address the evolving nature of phishing attacks and provide robust protection against cyber threats.



This is the base idea behind the proposed system. The code implementation is provided in the module section of this paper. The major steps involved are email inputting then there is feature extraction which is similar to that of the existing systems. Later we implement text tokenization and sequence padding where the text is tokenized and only the top most repeated words are considered for the convolution layer. Next we take a phishing email repository which can be found online (details provided in the reference section) our system will employ a deep learning architecture, specifically a CNN, designed to analyze email content comprehensively. CNNs excel in image and text analysis and have demonstrated remarkable capabilities in feature extraction and pattern recognition. The CNN will be configured to automatically extract relevant features from email text, including linguistic patterns, structural characteristics, and textual cues indicative of phishing attempts. In addition to email text, the proposed system may incorporate multiple input channels, such as email headers, sender information, and metadata, allowing for a holistic analysis of emails. Our system will provide real-time monitoring capabilities, enabling it to scan incoming emails for potential phishing threats as they arrive in the inbox. We may explore the use of ensemble learning techniques, combining the predictions of multiple models to improve accuracy and reduce false positives. An intuitive and user-friendly interface will be developed to allow users to interact with the system, report suspicious emails, and customize detection settings.

IX. ADVANTAGES OF PROPOSED SYSTEM

A. *Deep Learning Architecture*

Our system will employ a deep learning architecture, specifically a CNN, designed to analyze email content comprehensively. CNNs excel in image and text analysis and have demonstrated remarkable capabilities in feature extraction and pattern recognition.

B. *Multiple Input Channels*

In addition to email text, the proposed system may incorporate multiple input channels, such as email headers, sender information, and metadata, allowing for a holistic analysis of emails.

C. *User-Friendly Interface*

An intuitive and user-friendly interface will be developed to allow users to interact with the system, report suspicious emails, and customize detection settings.

D. *Enhanced Accuracy*

The deep learning architecture, combined with feature extraction capabilities, is expected to significantly enhance the accuracy of phishing email detection.

E. *Real-Time Protection*

By offering real-time monitoring, our system can swiftly identify and respond to phishing threats, reducing the risk of successful attacks.

F. *Adaptability*

Regular model updates ensure that the system remains effective against evolving phishing techniques and tactics.

G. *Holistic Analysis*

Multiple input channels and comprehensive feature extraction enable a holistic analysis of emails, improving detection capabilities.

H. *Ensemble Learning*

We may explore the use of ensemble learning techniques, combining the predictions of multiple models to improve accuracy and reduce false positives.

I. *Regular Model Updating*

To keep the system current and adaptive to emerging threats, regular model updates will be scheduled. These updates will incorporate the latest data and threat intelligence.

The proposed phishing email detection system represents a significant advancement in email security. By leveraging deep learning, real-time monitoring, and ensemble techniques, we aim to provide robust protection against phishing threats.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue X Oct 2023- Available at www.ijraset.com

As we move forward with the development and implementation of this system, our commitment to staying at the forefront of cybersecurity remains unwavering, with the ultimate goal of safeguarding individuals and organizations from the ever-present dangers of phishing attacks.

X. IMPLEMENTATION OF THE PROPOSED SYSTEM

A. System Requirements

The successful implementation of the phishing email detection system necessitates the following requirements:

1) *Hardware:* Adequate computational resources, such as a machine with a GPU or access to cloud-based GPU resources[8], to facilitate the training of the deep learning model efficiently.
2) *Software:* The system requires the following software libraries and tools:
   -Python 3.x
   -TensorFlow and Keras for deep learning model development
   -NumPy for numerical operations
   -pandas for data handling
   -Matplotlib or other suitable libraries for data visualization
   -A dataset of labeled phishing and legitimate emails (explained below)

B. Dataset

In this implementation, we load a sample phishing email dataset from the 'phishing_emails.csv' file. This dataset includes columns for 'email_text' containing the email content and 'is_phishing' indicating whether the email is a phishing attempt (1) or not (0).

3	Subject: allegheny energy s - 3 i received word from mike morrell at allegheny today that no sec review of the s - 3 will be required . obviously , one less potential delay to have to deal with . thanks , don	0
4	The University of Washington System is sharing funds for all students during this pandemic, please update your financial aid status to claim yours. Login.uw.edu/covid-19-aid-update For instructions on Accepting Your Financial Aid on https://login.uw.edu/login/login./. Regards, Assistant Professor University of Washington	1

The complete dataset can be downloaded from the link provided in the reference section. The given dataset is comparatively a smaller one which does not require a large GPU for running the python module. A good computer with minimum specifications can run the code.

C. Code

```
# Import necessary libraries
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Dense, Dropout, Embedding,Conv1D, GlobalMaxPooling1D
from tensorflow.keras.models import Sequential
```

```
# Load the phishing email dataset
data = pd.read_csv('phishing_emails.csv',encoding= 'unicode_escape')
# Preprocessing the data
tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(data['email_text'])
X = tokenizer.texts_to_sequences(data['email_text'])
X = pad_sequences(X, maxlen=500)
# Create the CNN model
model = Sequential()
model.add(Embedding(input_dim=10000, output_dim=64, input_length=500))
model.add(Conv1D(filters=64, kernel_size=5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dense(units=64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(units=1, activation='sigmoid'))
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X, data['is_phishing'], epochs=10, batch_size=32, validation_split=0.2)
# Evaluate the model's performance
test_loss, test_acc = model.evaluate(X, data['is_phishing'])
print('Test accuracy:', test_acc)
# Detect if a given email is a phishing email
new_email = 'Dear customer, your account has been compromised. Please click the link to reset your password.'
new_email_sequence = tokenizer.texts_to_sequences([new_email])
new_email_padded = pad_sequences(new_email_sequence, maxlen=500)
prediction = model.predict(new_email_padded)
# Train the model
if prediction > 0.5:
print('The email is a phishing email.')
else:
print('The email is not a phishing email.')
```

1) Importing Libraries

The code begins by importing necessary Python libraries, including NumPy, Pandas, and TensorFlow's Keras API for building deep learning models.

2) Loading the Dataset

It reads a dataset of phishing emails from a CSV file called 'phishing_emails.csv' using Pandas. The dataset contains email text and labels indicating whether each email is a phishing email or not.

3) Data Preprocessing

Text data preprocessing is performed to prepare the email text for input into the model. The Tokenizer class from Keras is used to convert the text data into sequences of integers and create a word vocabulary. num_words is set to 10,000, which means the tokenizer will consider the top 10,000 most frequent words in the dataset. The texts_to_sequences method converts each email text into a sequence of integer tokens. The pad sequences method ensures that all sequences have the same length by padding or truncating them to a maximum length of 500.

4) Creating the CNN Model

A sequential Keras model is created to build the CNN. An embedding layer is added to convert the integer-encoded tokens into dense vectors.

The input dimension is set to 10,000 (the vocabulary size), and the output dimension is 64.A 1D convolutional layer (Conv1D) is added with 64 filters and a kernel size of 5. It uses the ReLU activation function.A global max-pooling layer (GlobalMaxPooling1D) is added to extract the most important features from the convolutional layer.A dense layer with 64 units and ReLU activation is added followed by a dropout layer with a dropout rate of 50% to prevent overfitting. The final output layer consists of one unit with sigmoid activation, making it suitable for binary classification (phishing or not phishing).

5) Model Compilation

The model is compiled using the Adam optimizer and binary cross-entropy loss, which is appropriate for binary classification. The model is configured to track accuracy as a metric.

6) Model Training

The model is trained using the email text data (X) and labels (data['is_phishing']) for 10 epochs with a batch size of 32. A validation split of 20% is used to monitor the model's performance during training.

7) Model Evaluation

The code evaluates the model's performance on the same dataset it was trained on. It calculates the test loss and test accuracy and prints the test accuracy.

8) Email Phishing Detection

Finally, the code demonstrates how to use the trained model to detect whether a given email is a phishing email. It takes a sample email text, tokenizes and pads it, and then makes a prediction using the model. If the predicted probability is greater than 0.5, it classifies the email as a phishing email; otherwise, it classifies it as not phishing.

XI. KEY RESULTS

After running the code with over 10 epoch. Here's a summary of the training process:

In the first epoch, the training accuracy was approximately 56.15%, and the validation accuracy was approximately 66.67%. The loss was around 0.6878 for training and 0.6776 for validation.

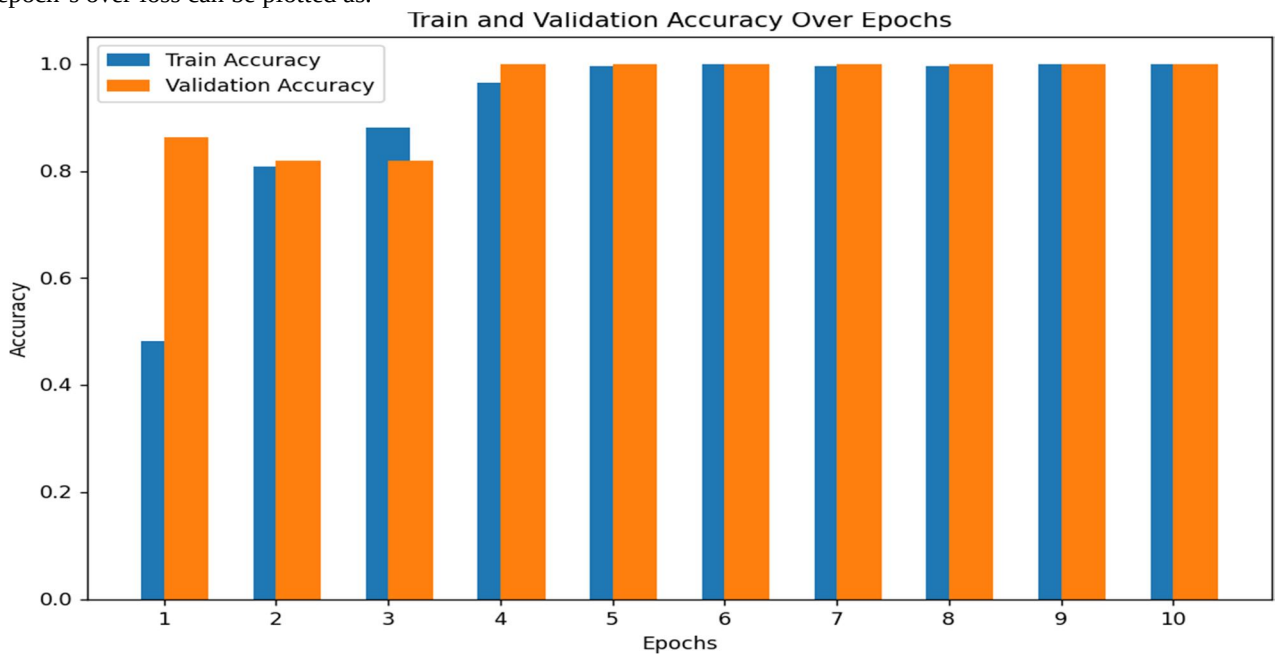
In the second epoch, the training accuracy increased to approximately 75.00%, and the validation accuracy improved to approximately 78.79%. The loss decreased to around 0.6501 for training and 0.6537 for validation.

In the third epoch, the training accuracy continued to improve to approximately 90.00%, and the validation accuracy reached 100.00%. The loss decreased further to around 0.6042 for training and 0.6142 for validation.

In the following epochs (4 to 10), the training accuracy remained high, reaching 100.00%, and the validation accuracy was also consistently 100.00%. The loss continued to decrease.

```
Epoch 1/10
9/9 [==============================] - 1s 49ms/step - loss: 0.6935 - accuracy: 0.4808 - val_loss: 0.6845 - val_accuracy: 0.8636
Epoch 2/10
9/9 [==============================] - 0s 33ms/step - loss: 0.6613 - accuracy: 0.8077 - val_loss: 0.6637 - val_accuracy: 0.8182
Epoch 3/10
9/9 [==============================] - 0s 31ms/step - loss: 0.6154 - accuracy: 0.8808 - val_loss: 0.6320 - val_accuracy: 0.8182
Epoch 4/10
9/9 [==============================] - 0s 31ms/step - loss: 0.5488 - accuracy: 0.9654 - val_loss: 0.5769 - val_accuracy: 1.0000
Epoch 5/10
9/9 [==============================] - 0s 32ms/step - loss: 0.4696 - accuracy: 0.9962 - val_loss: 0.4900 - val_accuracy: 1.0000
Epoch 6/10
9/9 [==============================] - 0s 32ms/step - loss: 0.3581 - accuracy: 1.0000 - val_loss: 0.3639 - val_accuracy: 1.0000
Epoch 7/10
9/9 [==============================] - 0s 33ms/step - loss: 0.2321 - accuracy: 0.9962 - val_loss: 0.2282 - val_accuracy: 1.0000
Epoch 8/10
9/9 [==============================] - 0s 35ms/step - loss: 0.1245 - accuracy: 0.9962 - val_loss: 0.1130 - val_accuracy: 1.0000
Epoch 9/10
9/9 [==============================] - 0s 34ms/step - loss: 0.0585 - accuracy: 1.0000 - val_loss: 0.0515 - val_accuracy: 1.0000
Epoch 10/10
9/9 [==============================] - 0s 32ms/step - loss: 0.0327 - accuracy: 1.0000 - val_loss: 0.0254 - val_accuracy: 1.0000
11/11 [==============================] - 0s 5ms/step - loss: 0.0152 - accuracy: 1.0000
Test accuracy: 1.0
1/1 [==============================] - 0s 70ms/step
The email is a phishing email.
```

The epoch's over loss can be plotted as:



XII. FUTURE ADVANCEMENTS

While the proposed system holds great promise, several key areas of Future enhancements which we personally think would benefit are:

A. Expansion of Datasets

1) *Quantity:* Increasing the scale of the training dataset is vital. A large and greater diverse dataset can assist enhance model generalization and robustness.
2) *Diverse Sources:* Diversifying the resources of the dataset by way of along with emails from diverse industries, regions, and email customers can better simulate real-global conditions.
3) *Balanced Distribution:* Ensuring a balanced distribution of phishing and valid emails inside the dataset enables the version keep away from biases and perform greater successfully across both classes.

B. Multilingual Support

1) *Language Diversity:* Extending language aid beyond English to embody a big range of languages is essential in addressing international phishing threats.
2) *Multilingual Training:* Training the version to come across phishing attempts in multiple languages, with appropriate tokenization and embedding techniques, complements its applicability in multicultural contexts.

C. Variety of Emails

1) *Email Types:* Expanding the scope of the system to investigate exclusive kinds of emails, which includes promotional emails, newsletters, and transactional emails, can offer a greater nuanced expertise of e mail content material.

2) *Rich Media:* Incorporating aid for emails with rich media content material, together with pictures, attachments, and embedded hyperlinks, can provide greater complete analysis abilities.

D. Enhanced Model Architecture

1) *Ensemble Models:* Exploring ensemble getting to know strategies via combining the predictions of a couple of models can further enhance accuracy and reduce false positives.
2) *Recurrent Architectures:* Considering the integration of recurrent neural community (RNN) additives along CNNs can help seize sequential patterns inside emails greater successfully.

E. Continuous Research and Collaboration

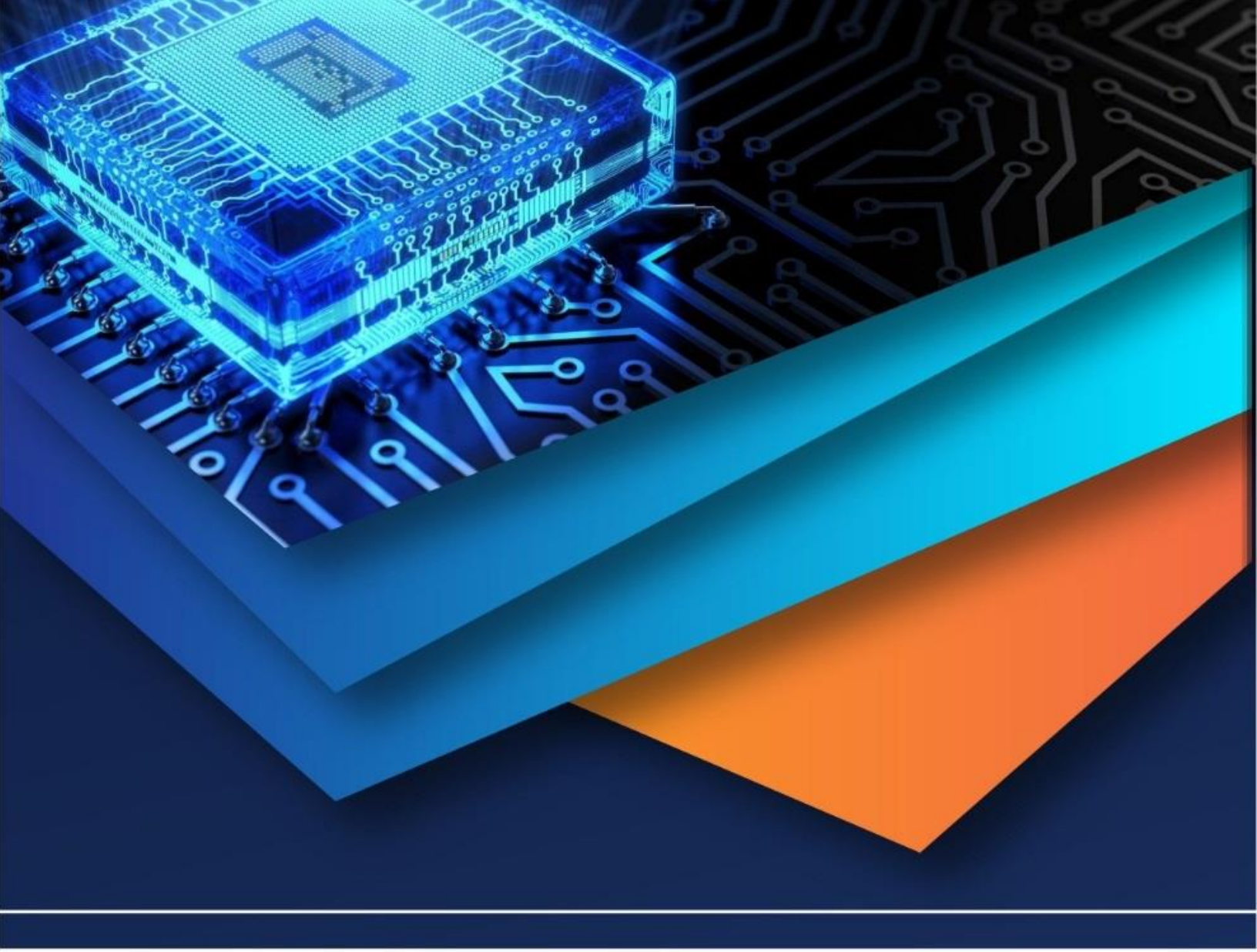
1) *Collaboration:* Engaging in collaborative research efforts with academia, cybersecurity organizations, and industry partners to stay at the forefront of phishing threat intelligence.
2) *Threat Intelligence Integration:* Incorporating external threat intelligence feeds and APIs to enhance the system's threat detection capabilities.

XIII. CONCLUSION

In an ever-changing cybersecurity landscape, combating phishing email attacks is a daunting challenge. This paper presented a detailed study of phishing email detection using the capabilities of Convolutional Neural Networks (CNNs). Through an in-depth analysis of existing systems, a solid foundation has been laid for an improved system that can reduce the risks associated with attempted arrests for abuse. The proposed system combines cutting-edge deep learning techniques with real-time monitoring, adaptability, and user-friendliness, offering a multifaceted approach to the detection of phishing emails. As we finish this paper, it's far vital to emphasize that while generation performs a pivotal role inside the combat in opposition to cyber threats, user attention, training, and vigilance are equally critical. Together, via an aggregate of modern answers and knowledgeable practices, we will create an impressive defense against phishing emails, fortifying the safety of our digital international.

REFERENCES

[1] “A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN” by Zainab Alshingiti 1,Rabeah Alaqel 1,Jalal Al-Muhtadi 1,2,Qazi Emad Ul Haq 3,*,Kashif Saleem 2ORCID and Muhammad Hamza Faheem 3ORCID
[2] “Convolutional Neural Network Optimization for Phishing Email Classification” by Cameron McGinley; Sergio A. Salinas Monroy
[3] https://www.statista.com/topics/8385/phishing/
[4] https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/#:~:text=LSTM%20(Long%20Short%2DTerm%20Memory,ideal%20for%20sequence%20prediction%20tasks.
[5] https://www.tensorflow.org/api_docs



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY