WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Phishing Detection Model Using Recurrent Neural Network in Web-Based Services

Joseph Ebieteli Perfect[a], Matthew Ehikhamenle[b]

[a] PG Student, Centre for Information and Telecommunication Engineering, Faculty of Engineering, University of Port Harcourt, Rivers State, Nigeria.
[b] Senior Lecturer, Department of Electrical and Electronic Engineering, University of Port Harcourt, Rivers State, Nigeria

ABSTRACT:

Phishing attacks present a notable risk to online activities by taking advantage of the trust and vulnerability of an individual to stealthily obtain sensitive information. Recurrent Neural Networks (RNNs) are recognized as promising tools for identifying phishing attacks due to their capacity to capture temporal relationships and context in sequential data. This report explores the process of phishing detection, with a focus on building and assessing a Custom Recurrent Neural Network (CRNN) model. The CRNN model is carefully designed to improve the precision and effectiveness of identifying phishing websites through the analysis of subtle and complex patterns in URLs and Emails. Exploratory Data Analysis (EDA) was vital in comprehending the structure of the dataset used and in shaping the data preprocessing approaches. Subsequently, an extensive investigation on the effectiveness of phishing detection using advanced machine learning models, such as the Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU), and a custom LSTM (CRNN) model was executed by training and testing the dataset with the aforementioned models. The custom LSTM model had the highest Accuracy score of 92.6%, followed by the highest F1 score of 92.5% and the least Test Loss score of 20.3% as against the default LSTM Model with relative scores of 91.8%, 91.6% and 21.7% respectively. The results indicate that the Custom Recurrent Neural Network (CRNN) model exhibits superior outcomes in identifying phishing websites based on customizing the attention mechanism structure of the default LSTM model, which is directly proportionate to the strength and adaptability in achieving an enhanced detection mechanism.

Keywords: Recurrent Neural network (RNN), Long Short-Term Memory (LSTM), Uniform Resource Locator (URL), Custom Recurrent Neural Network (CRNN), Gated Recurrent Network (GRU), Bi-Directional LSTM (Bi-LSTM), Exploratory Data Analysis (EDA)

Introduction:

The proliferation of Web-based Services has revolutionized communication, enabling seamless connectivity and access to information. However, this connectivity also introduces security risks, with phishing emerging as a prevalent threat. Detecting and mitigating phishing attacks is paramount to safeguarding users and maintaining trust in digital communication systems.

Traditional phishing detection methods often rely on static rules, blacklists, or heuristics, which may struggle to adapt to evolving phishing tactics and sophisticated attack techniques. Moreover, the dynamic and heterogeneous nature of Web Based Services poses additional challenges for effective phishing detection. Conventional approaches may be ill-suited to handle the complexities of wireless communication channels, including varying signal strengths, noise levels, and transmission protocols.

Previous research has explored various machine learning and data mining techniques for phishing detection, including decision trees, support vector machines (SVMs), and ensemble methods.

The use of RNNs for sequence-based modeling has shown promising results in other domains, suggesting their potential applicability for phishing detection in web-based services.

Literature Review.

Phishing is a kind of hack where cybercriminals pose as reliable organizations to steal confidential data. In the current technological era, phishing has become a major threat. This risk has been exacerbated by the growing reliance on internet-based services and the quick development of wireless connectivity, which has led to an increase in the frequency of phishing scams and their evolution into more complex and widespread forms.

Phishing is a criminal activity that uses technical deception as well as social engineering to get the personal information and bank account credentials of victims. Social engineering methods take advantage of gullible people by using tricks like phony email addresses and communications to trick them into thinking they are communicating with a reliable and authentic source. These are intended to direct customers to fraudulent websites that entice recipients

to reveal financial information, including passwords and usernames. Technically, deception methods involve systems that intercept users' usernames and passwords or divert users to phony websites in order to plant malware on computers and collect credentials directly. APWG, (2023).

In a report recently released by the Anti-Phishing Working Group (APWG), the number of phishing attacks that transpired in the third quarter of 2023 is alarmingly reported. The report reveals that a staggering 792,712 phishing attacks took place during this period, which is a significant increase compared to previous years. This surge in phishing incidents clearly indicates a rise in the sophistication and prevalence of these malicious tactics employed by cybercriminals.

The financial incentive that drives phishing attacks is both unequivocal and profound, as expressed in a report from Cybersecurity Ventures, where it is projected that the global costs of cybercrime will skyrocket to an astonishing $10.5 trillion by the year 2025. This staggering figure serves as a testament to the immense profitability and allure that illicit activities such as phishing possess. Given the substantial financial potential that is at stake, it becomes clear that cybercriminals are highly motivated to continuously refine and escalate their tactics, resulting in the proliferation of increasingly sophisticated and elaborate phishing campaigns. Morgan, Steve, (2022).

The financial consequences of cybercrime extend far beyond the immediate gains that are acquired through successful phishing attacks. Alongside the direct monetary losses that victims experience due to fraudulent transactions or unauthorized access to their financial accounts, cybercrime also imposes significant indirect costs on businesses, governments, and society as a whole. These costs may include various expenses related to incident response, forensic investigations, legal proceedings, regulatory fines, reputational damage, as well as the loss of customer trust. Moreover, the ripple effects of cybercrime have the potential to disrupt critical infrastructure, undermine economic stability, and erode public confidence in digital technologies.

The projected escalation of global cybercrime costs serves as a powerful incentive for cybercriminals to invest in the development of increasingly sophisticated phishing campaigns. By leveraging advanced social engineering techniques, exploiting vulnerabilities in software and hardware, and employing deceptive tactics to evade detection, attackers are able to maximize their chances of achieving success and reaping substantial financial rewards. Furthermore, the anonymous and borderless nature of cyberspace provides cybercriminals with the ability to operate with relative impunity, further incentivizing their engagement in illicit activities.

Phishers have refined their craft by utilizing a wide range of psychological tactics that take advantage of human vulnerabilities and expertly capitalize on cognitive biases and emotional triggers to manipulate the decision-making processes of unsuspecting individuals. Phishing emails, texts, and phone calls are expertly constructed to elicit an immediate response from their recipients, often managing to evade traditional security defenses and detection mechanisms. The world of phishing campaigns has evolved significantly in terms of sophistication and adaptability in recent years, posing formidable challenges to cybersecurity practitioners and researchers alike. Ross, (2016)

Related works on Artificial Neural Network and Phishing Detection?

S.S. Roy, A.I. Awad, L.A. Amare, M.T. Erkihun, and M. Anas. (2022)) in A Multimodal Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU Models highlighted the effectiveness of these deep learning models in detecting phishing websites and emphasized the importance of using advanced techniques to combat cyber threats. Although, there were limited Dataset, lack of real-time evaluation, interpretability of models, and scalability issues emerged.

Maruf Ahmed Tamal, Md Kabirul Islam, Touhid Bhuiyan, and Abdus Sattar. (2024) in Dataset of Suspicious Phishing URL Detection findings included the development of a large-scale labeled dataset specifically designed for URL-based phishing detection. Although some of the limitations included a lack of novelty in the approach to dataset preparation despite the innovation with 10 novel features. The methodology used in creating the dataset aligns with common practices seen in similar existing datasets, suggesting a need for exploring alternative approaches to enhance originality. APWG (Anti-Phishing Working Group) November 2023 in Phishing Activity Trends Report 1st Quarter 2023 reported that phishing attacks reached a record high in early 2023, the report also measures the evolution,

proliferation, and propagation of identity theft methods. Sunil Chaudhary (2012) in Recognition of Phishing Attacks Utilizing Anomalies in Phishing Websites, concluded the identification of forty-one anomalies in URLs and source codes of phishing websites through meta-analysis of existing phishing prevention techniques. He discovered that some anomalies previously significant for phishing detection are no longer prevalent in current phishing websites, while certain anomalies are also commonly found in legitimate websites. It was later discovered that the dynamic nature of phishing attacks and techniques may impact the relevance of identified anomalies over time. Continuous monitoring and updating of anomalies are essential to ensure the effectiveness of phishing detection methods.

Nemat Ullah Noori, et al, in Phishing URL Detection using Machine Learning, concluded results in his study included the evaluation of the SVM classifier's performance for different sizes of datasets using various numbers of features. The study compares the results with other machine learning classification techniques and highlights the effectiveness of the proposed system in detecting phishing websites using URL features only.

Methodology:

Phishing attacks stand as a persistent and substantial threat to online security, exploiting the trust and vulnerability of users to surreptitiously extract sensitive information. These malicious tactics, often cloaked in legitimate-looking emails or websites, can result in dire consequences ranging from financial loss to identity theft. As the sophistication of these attacks continues to evolve, the need for robust and efficient detection mechanisms becomes increasingly imperative.

In recent years, Recurrent Neural Networks (RNNs) have emerged as a promising tool in the realm of phishing detection. RNNs are particularly well-suited for tasks involving sequential data, such as text, making them a natural fit for analysing the intricate patterns within URLs and webpage content. By capturing the temporal dependencies and context within these sequences, RNNs have demonstrated their potential to effectively discern between legitimate and phishing URLs.

Building upon this foundation, this research proposes an innovative approach, A Custom Recurrent Neural Network model tailored specifically for the nuanced challenges of phishing detection. The aim of this model is to not only enhance the accuracy of phishing detection but also to improve the efficiency of the overall detection process.

By leveraging the strengths of RNNs while integrating novel enhancements, the CRNN model seeks to elevate the capabilities of current phishing detection systems.

One of the key advantages of the proposed CRNN model lies in its ability to delve deep into the structural and content-related features of URLs. By incorporating attention mechanisms within the RNN architecture, the model can selectively focus on critical segments of URLs, potentially revealing subtle indicators of phishing attempts that might evade traditional detection methods. This targeted attention allows the model to hone in on suspicious patterns, thus increasing its discriminatory power between legitimate and malicious URLs. Furthermore, the CRNN model will be trained on a comprehensive dataset that encompasses a wide array of URL attributes, ranging from URL length and domain age to the presence of HTTPS encryption.

Through rigorous training and validation, the model will learn to discern the intricate patterns that distinguish phishing URLs from legitimate ones. The inclusion of attention mechanisms not only enhances the model's performance but also provides valuable interpretability, allowing analysts to understand the specific cues that contribute to a classification decision

Research Objectives:

The central focus of this research endeavour revolves around the primary objective of designing and meticulously evaluating a Custom Recurrent Neural Network model.

This model is meticulously crafted with the explicit aim of advancing the current landscape of phishing website detection methodologies. The overarching goal is not only to enhance the accuracy but also to significantly bolster the efficiency of detecting these insidious phishing attempts, thereby setting a new benchmark for comparison against traditional detection methods.

Furthermore, we commenced by acquiring the dataset and preprocessing it to handle any missing values and extract relevant features. Next, we develop multiple models, including baseline models like LSTM, GRU, and Bidirectional LSTM, as well as a custom model. These models are then trained on the training dataset and evaluated on the testing dataset to assess their performance. Based on the evaluation results, the best-performing model is selected for deployment in a production environment. Finally, the deployed model undergoes continuous improvement through monitoring and refinement to adapt to changing phishing threats and ensure ongoing effectiveness.

Moreover, the content-related attributes in the dataset furnish the CRNN model with a sophisticated comprehension of the textual and structural components embedded in URLs. These attributes encompass various factors such as the presence of specific keywords that indicate phishing, employment of JavaScript redirection, and existence of pop-up windows—all of which are indicative of potential phishing endeavours. By exposing the model to this diverse set of features, the dataset enables it to navigate the complex realm of phishing detection with accuracy and insight. The Phishing Websites dataset goes beyond being a static reservoir of information; it represents a dynamic canvas reflecting the evolving landscape of cyber threats. As new phishing methodologies and tactics surface, the dataset can be updated and expanded to integrate these advancements, ensuring that the CRNN model retains its relevance and efficacy in the midst of constantly changing threats.

URL Data Repository

For the purpose of conducting this research endeavour, it is of utmost importance to delve deeply into the vast reservoir of publicly accessible datasets that function as highly valuable assets for the purpose of both training and assessing the efficacy of machine learning models. One particularly suitable candidate for examination within the scope of this study is the Phishing Websites dataset, a dataset of significant repute and widespread utilization that can be readily accessed through popular platforms such as Kaggle.

This dataset stands out as a veritable goldmine of indispensable information, meticulously compiled to encompass a wide-ranging assortment of features that are highly relevant to the domain of detecting and combatting phishing activities.

The Phishing Websites dataset, at its essence, presents an extensive array of attributes that encapsulate the intricate nature of phishing attempts comprehensively.

These attributes cover a broad spectrum, ranging from the fundamental features of URLs to the detailed aspects of domain information and content-related attributes. Each entry in this dataset represents a carefully constructed snapshot, capturing the subtleties and nuances that differentiate authentic websites from deceptive phishing sites.

The incorporation of features related to URLs in the dataset acts as a fundamental cornerstone for the training and assessment of the CRNN model. Even seemingly trivial attributes like URL length can provide crucial insights into the authenticity of a website. Similarly, the presence or absence of HTTPS

encryption, a vital indicator of secure communication, is a feature that the model will learn to examine meticulously. Additionally, the dataset offers a diverse range of domain information, including details on domain age and registrar, which provide valuable clues for detecting malicious intent.

Methods of Data Analysis:

The preprocessing phase plays a vital role as a necessary precursor to model development, encompassing a series of meticulous procedures to guarantee the quality and integrity of the dataset. Within these foundational processes, data cleaning emerges as the initial step, with the aim of eliminating redundancies and rectifying missing values within the dataset. This data cleaning procedure is not just a routine task but a crucial effort that establishes the basis for precise and dependable analysis. The eradication of duplicates is a fundamental aspect of data cleaning, addressing the potential bias that duplicate entries may introduce to the dataset. By systematically detecting and removing these duplicate records, the dataset is cleansed of redundancy, ensuring that each data point contributes uniquely to the subsequent analysis. Concurrently, managing missing values represents another essential aspect of data cleaning. Unaddressed missing values can distort the outcomes of later analyses, leading to inaccurate conclusions. Through meticulous methods like imputation or deletion, missing values are carefully handled, thereby enhancing the dataset's comprehensiveness and dependability. Following the data cleansing and purification procedure, the trip continues to feature engineering, a skilled technique that involves extracting and creating new features from existing ones. Feature engineering is important in phishing detection because it allows the model to discern tiny changes between valid and fake URLs. An exemplary instance of feature engineering in this domain includes extracting pertinent features like URL length, domain age, and SSL encryption presence. For example, URL length can be a powerful indicator, as phishing URLs often have distinct lengths compared to legitimate ones. Similarly, domain age and SSL encryption presence can provide valuable insights into a website's credibility and security posture. By extracting and integrating these features, the model gains a detailed understanding of the attributes defining phishing attempts.

Furthermore, tokenization emerges as a crucial step in the feature engineering process, especially when working with textual data such as URLs. Tokenization involves converting raw text data into numerical tokens, enabling the model to process textual information and extract meaningful patterns and structures within URLs. As URLs are tokenized into numerical forms, the model acquires the capability to identify intricate patterns and relationships embedded in these textual strings. This transformation not only enhances the model's ability to process textual data but also allows it to capture the sequential nature of URLs, which is essential for tasks involving sequential data analysis.

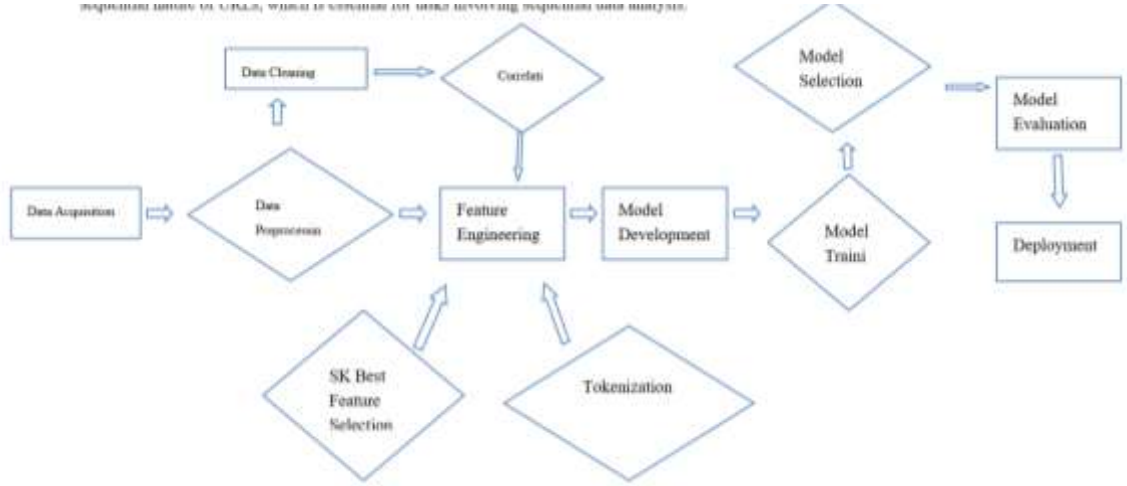

Fig 1 Block Diagram

Exploratory Data Analysis:

Exploratory Data Analysis (EDA) is a crucial phase, providing insights into the underlying structure and characteristics of the dataset. In the context of our research on phishing website detection, EDA played a pivotal role in understanding the composition of the dataset and identifying any necessary adjustments to ensure its suitability for analysis. First and foremost, it was imperative to gain an understanding of the size and dimensions of the dataset. With a dataset of considerable size containing numerous features, it was essential to assess its overall structure and complexity. This allowed for informed decision-making regarding data preprocessing and modelling strategies. The Phishing Websites dataset comprises a total of 89 features, each meticulously selected to encapsulate various aspects related to URLs, domain characteristics, and content-related attributes. These features offer a comprehensive view of potential indicators of phishing attempts, ranging from basic URL attributes to intricate domain and content-based characteristics. The features within the dataset exhibit a diverse range of structures and natures, spanning from binary indicators (e.g., presence of HTTPS encryption) to numerical attributes (e.g., domain registration length). Additionally, textual features, such as the presence of specific keywords or phrases, contribute to the multifaceted nature of the dataset.

Data Cleaning:

Data cleaning is an essential step in any data analysis or machine learning project, ensuring the integrity and reliability of the dataset. In the context of cybersecurity research, where accurate detection of phishing websites is crucial, data cleaning plays a critical role in preparing the dataset for analysis. The process of data cleaning involves identifying and addressing missing values and duplicate records in the dataset. Missing values can distort analysis

results and introduce bias, while duplicate records can skew statistical measures and lead to erroneous conclusions. By systematically removing missing values and duplicate records, researchers can ensure the accuracy and validity of their analyses. Fortunately, in the case of the phishing website dataset used in this research, the data was clean and free from missing values or duplicate records. As a result, there was no need to go through the process of data cleaning. This allowed for a more streamlined analysis and ensured that the insights derived from the dataset were accurate and reliable. While data cleaning may not always be necessary, it remains an essential aspect of data analysis and machine learning, particularly in datasets collected from diverse sources which are prone to errors. By maintaining data cleanliness, researchers can trust the integrity of their analyses and make informed decisions based on reliable data.

Feature Engineering:

Feature engineering is a pivotal step in the machine learning pipeline, essential for transforming raw data into informative features that facilitate accurate model predictions. In our exploration of phishing website detection, several feature engineering techniques were employed to extract meaningful insights from the dataset and construct informative input variables for our models. Certain columns in the dataset, deemed irrelevant or redundant for the task of phishing detection, were dropped to streamline the feature space and focus on the most relevant attributes. Columns such as 'url', 'ip', 'punycode', and others were excluded from the analysis to remove noise and reduce dimensionality. To prepare the dataset for training, we should drop columns that are not relevant or may not contribute significantly to the prediction task. They are domain specific or not expected to provide meaningful information. Here's a list of columns that were

considered for dropping, the target variable, indicating whether a website is legitimate or phishing, was extracted from the dataset to serve as the label for supervised learning. This binary classification task formed the basis of our predictive modelling efforts, with the goal of accurately classifying websites based on their characteristics. The variable y was defined to hold the target “status”.

```
y = data_frame["status"]
```

By extracting the target variable from the dataset, we establish a clear framework for training our predictive model. The model learns to identify patterns and relationships in the input features that are indicative of the target label, enabling it to make accurate predictions on unseen data. The target variable serves as the basis for evaluating the performance of the predictive model. During training, the model's predictions are compared to the actual target values, allowing us to assess its accuracy, precision, recall, and other performance metrics. This evaluation process is essential for gauging the model's effectiveness in correctly classifying websites and identifying potential phishing threats.

Data Split:

The dataset was partitioned into training and testing sets to facilitate model training and evaluation. This ensured that the model's performance could be assessed on unseen data, providing a reliable estimate of its generalization capabilities. The dataset splitting process involves dividing the available data into two or more subsets, typically referred to as the training set and the testing set. The training set is used to train the model, while the testing set is used to evaluate its performance.

The dataset splitting method ensures that the model's performance is evaluated on data that it has not previously been exposed to for dataset splitting, we used Python's scikit-learn module, which has a variety of machine learning techniques, including dataset splitting functionality. We utilized the 'train_test_split' method from the'sklearn.model_selection' module to divide the dataset into training and testing sets. This function enables customizable splitting based on user-specified criteria such as test size, random state, and stratification.ways to carry out training The train_test_split function works by randomly partitioning the dataset into two subsets according to the specified test size ratio. By default, it shuffles the data before splitting to ensure that the distribution of classes or labels is preserved in both subsets. Additionally, the function supports stratified splitting, which ensures that the proportion of classes or labels is maintained in both the training and testing sets, particularly useful for imbalanced datasets. Splitting the dataset allows us to evaluate the model's performance on unseen data, providing a more accurate estimate of its effectiveness in real-world scenarios. This helps prevent overfitting, where the model memorizes the training data without generalizing well to new observations. Dataset splitting facilitates hyperparameter tuning by enabling the evaluation of different model configurations on the testing set. This helps identify the optimal hyperparameters that maximize the model's performance. Dataset splitting allows for fair comparisons between different models or algorithms by evaluating their performance on the same testing set. This helps in selecting the best performing model for the given task.

Data Nominalization:

Normalization is a crucial preprocessing step in machine learning that involves scaling numerical features to a standard range, typically between 0 and 1 or with a mean of 0 and a standard deviation of 1. Normalization is essential for ensuring that numerical features contribute equally to the learning process, regardless of their original scale or magnitude. Without normalization, features with larger magnitudes may dominate the learning process and overshadow the contributions of other features, leading to biased model predictions and suboptimal performance.

In our phishing detection project, normalization was applied to ensure that all numerical features in the dataset were on a consistent scale. This was particularly important given the diverse range of features, each capturing different aspects of URLs and website characteristics. By normalizing the features, we aimed to mitigate the potential impact of varying scales and magnitudes on the performance of our machine learning models.

For normalization, we leveraged the ‘StandardScaler’ class from the scikit-learn library in Python.

Scikit-learn is a powerful and widely used machine learning library that provides a comprehensive set of tools for various tasks, including preprocessing, model training, and evaluation. The 'StandardScaler' class implements normalization by subtracting the mean and dividing by the standard deviation of each feature, ensuring that the transformed features have a mean of 0 and a standard deviation of 1.

Feature selection is an important part of machine learning that entails identifying and selecting the most relevant features from a dataset in order to improve model performance and minimize overfitting. SelectKBest is a popular feature selection technique that aims to select the top k features based on their scores from a specified scoring function. We chose SelectKBest as our feature selection method due to its simplicity and effectiveness in identifying the most informative features from a dataset. By focusing on the top k features with the highest scores, SelectKBest allowed us to streamline our feature set and improve model performance. The first step in the SelectKBest method is to compute scores for each feature based on a specified scoring function. The scoring function evaluates the importance or relevance of each feature with respect to the target variable. Common scoring functions include;

1. Chi-Squared Test: Measures the dependence between categorical variables.
2. ANOVA F-value: Assesses the difference in means between groups for continuous variables.
3. Mutual Information: Measures the amount of information gained about one variable through another variable
4. Correlation Coefficient: Quantifies the linear relationship between two variables.

We opted for the f_classif scoring function, which is suitable for classification tasks and evaluates the relevance of each feature by analyzing the variance between classes. This scoring function is well-suited for identifying features that are discriminative for distinguishing between different classes in a classification problem. The f_classif (ANOVA F-value) method is a statistical technique used for feature selection, particularly in classification tasks. It calculates the F-value of each feature by analyzing the variance between

classes and within classes.

The between-class variance, also known as the sum of squares between (SSB), measures the variation of feature values between different classes.

$$\boldsymbol{SSB} = \sum_{i=1}^{k} \boldsymbol{n_i}(\bar{\boldsymbol{x}}_{\boldsymbol{i}} - \bar{\boldsymbol{x}})^{\boldsymbol{2}} \quad [1]$$

From equation [1], K is the number of classes; $n_i$is the number of samples in class i; $\bar{x}_i$is the mean of the feature values in class i.

The within-class variance, or the sum of squares within (SSW), measures the variation of feature values within each class.

$$\boldsymbol{SSW} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\boldsymbol{x_{ij}} - \bar{\boldsymbol{x}}_{\boldsymbol{i}})^{\boldsymbol{2}} \quad [2]$$

Where x_{ij} is the j-th feature value in class I; $\bar{x}_i$ the mean of the feature values in class i.

The total sum of squares, or the total variance (SST), measures the total variation of feature values across all samples. It is computed as the sum of squares of deviations from the overall mean.

The total sum of squares, or the total variance (SST), measures the total variation of feature values across all samples. It is computed as the sum of squares of deviations from the overall mean.

$$\boldsymbol{SST} = \sum_{i=1}^{k} \sum_{j=1}^{n_I} (\boldsymbol{x_{j=1}} - \bar{\boldsymbol{x}})^{\boldsymbol{2}} \quad [3]$$

Where bar x is the overall mean of the feature values. Once we have the between-class variance (SSB) and within-

class variance (SSW), we can calculate the F-value for each feature. The F-value is given by:

$$\boldsymbol{F} = \frac{\boldsymbol{MSB}}{\boldsymbol{MSW}} \quad [4]$$

Where MSB is the mean square between; MSW is the mean square within.

The f_classif method evaluates the significance of each feature by comparing the variation between classes to the variation within classes. Features with higher F-values indicate greater discrimination between classes and are thus selected as more relevant for classification tasks. The top 30 features with the highest scores were selected for inclusion in the final feature set.

Model Architecture:

At the heart of the CRNN model's architecture is the deliberate inclusion of attention mechanisms, a sophisticated enhancement that enables the model to focus on the crucial components within URLs. These attention mechanisms function as a spotlight, guiding the model's attention towards the pivotal segments of URLs that contain valuable indicators of potential phishing endeavors. By selectively concentrating on these significant segments, the model can identify subtle patterns and irregularities that may evade detection by traditional methods

The foundational element of this intricately crafted architecture is the embedding layer, which plays a crucial role in converting tokenized URLs into dense numerical representations. This embedding procedure not only facilitates the model's processing of textual data but also equips it with the capability to capture the semantic connections and contextual intricacies present in URLs. By transforming raw textual data into a structured numerical framework, the embedding layer establishes the basis for subsequent layers to extract meaningful information.

Expanding on this foundation, the architecture incorporates multiple Recurrent Neural Network (RNN) layers, including variations like Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), selected specifically for their effectiveness in handling sequential data. These RNN layers serve as the memory of the neural network, proficient in capturing the temporal relationships and sequential patterns inherent in URLs. By iteratively processing sequences, the model develops a comprehensive comprehension of the sequential characteristics of URLs, enabling it to differentiate between phishing URLs and legitimate ones based on subtle distinctions.

Alongside the RNN layers, a critical aspect of the CRNN model's architecture is the attention layer, which acts as the perceptive component of the model, dynamically assigning attention to various segments of URLs based on their relevance to the given task. By concentrating on key URL segments such as domain names or specific path elements, the attention layer empowers the model to enhance the significance of crucial features while filtering out extraneous information. This selective attention mechanism not only improves the model's precision but also offers valuable interpretability, enabling analysts to comprehend the reasoning behind the model's determinations. Ultimately, the architecture is completed by fully connected layers, a traditional element of neural networks responsible for the final classification. These layers amalgamate the insights derived from the preceding stages, distilling them into a definitive classification of URLs as either phishing or legitimate. Through a sequence of nonlinear operations and activations, the fully connected layers guarantee that the model's output is not only precise but also confidently asserts its classification decisions.

Custom Model:

The custom LSTM model designed for the phishing detection project is aimed at improving the performance of the classification task by incorporating additional layers and regularization techniques.

Let X represent the input sequence with dimensions (T,n), where T denotes the number of time steps (sequence length) and n denotes the number of features.

The custom LSTM model consists of two LSTM layers followed by dropout layers for regularization and dense layers for classification. The first LSTM layer processes the input sequence X and produces a sequence of hidden states $H1$. Each LSTM unit in this layer comprises input, forget, and output gates along with a cell state, and the calculations are similar to those described in the previous message.

After the first LSTM layer, a dropout layer is applied to prevent overfitting. Dropout randomly sets a fraction of input units to zero during training, which helps in preventing the model from relying too much on specific features. The output of the first LSTM layer, $H1$, serves as the input sequence for the second LSTM layer. This layer processes the sequence and produces another sequence of hidden states $H2$. After the second LSTM layer, multiple dense layers are employed for classification. These layers apply linear transformations followed by activation functions to produce the final output probabilities. The final dense layer applies a linear transformation followed by the sigmoid activation function to produce the output probability for binary classification:

$$\hat{y} - \sigma\left(W_{out}H_t + b_{out}\right)$$
[5]

Where:

H_T denotes the final hidden state from the last LSTM layer.

W_{out} and b_{out} represent the weight matrix and bias for the output layer.

σ represents the sigmoid activation function.

The custom LSTM model incorporates additional layers and regularization techniques to enhance the model's ability to capture complex patterns in the input data and generalize well to unseen examples. By leveraging the expressive power of LSTM units and incorporating dropout regularization.

we developed a custom LSTM model within the TensorFlow framework to enhance the accuracy and robustness of our classification task. Let me provide an overview of how we implemented this custom model and its significance within the context of our project. The need for a custom model arose from the desire to improve the performance of our phishing detection system. While the original LSTM model showed promising results, we believed that a more sophisticated architecture could further enhance the model's ability to discern between legitimate and phishing websites. The custom LSTM model we designed consists of multiple layers, including LSTM layers, dropout layers for regularization, and dense layers for classification. This architecture allows the model to learn intricate patterns in the input data and make accurate predictions. We utilized LSTM layers to process sequential data, such as URL features extracted from websites. LSTM units within these layers enable the model to capture long-term dependencies in the input sequences. To prevent overfitting and improve generalization, we incorporated dropout layers after each LSTM layer. These layers randomly drop a fraction of input units during training, forcing the model to learn more robust representations. Following the LSTM layers, we included dense layers for classification. These layers apply linear transformations and activation functions to produce the final output probabilities, indicating the likelihood of a website being phishing or legitimate.

Results

The model development infrastructure includes the following

I. Platform: Google Collaboratory

II. Internal Memory: 13.3GB RAM

III. GPU: Tesla T4

IV. Storage: 106GB

V. Libraries used: Numpy, Pandas, Matplotlib, Seaborn, Scikit-Learn, and Scipy

VI. Data Augmentation: None

VII. Model Training: TensorFlow/Keras

For parameters:

I. Epoch: 50

II. Learning Rate:0.001

III. Optimizer: Adam

IV. IV. Dropout Rate: 0.2

Dataset Statistics:

A perfectly balanced dataset, with an equal number of legitimate and phishing instances (5715 each) was used and it provides an excellent foundation for model training. This balance minimizes model bias, simplifies the training process, and ensures that accuracy and other evaluation metrics reflect true model performance without skewing towards a majority class. Such a setup enhances the model's ability to generalize well to new, unseen data, significantly benefiting the deployment of robust, reliable phishing detection systems in real-world scenarios. This equilibrium in the dataset is ideal for straightforward model development and fair performance evaluations.

We implemented three key models, each utilizing a different architecture to leverage the sequential nature of the data. They are: LSTM, GRU, and the Bi-Direction

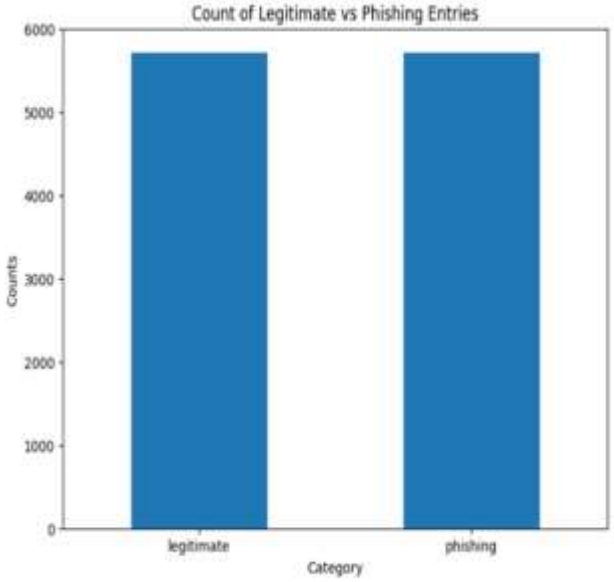

Figure 2: Distribution of Dataset for Training

Models

Four RNN model variation were employed. They are:

a. LSTM (Long Short-Term Memory)

b. GRU (Gated Recurrent Unit)

c. Bidirectional LSTM

d. Custom (User defined) Network architecture

Analysis of Performance Metrics Website Phishing Model

From the analysis, the Custom LSTM model demonstrated the best performance in terms of test accuracy, F1 score, and test loss. This highlights the importance of customizing and fine-tuning deep learning models for specific tasks. The LSTM and Bidirectional LSTM models also showed strong

performance, while the GRU model, despite being computationally efficient, was slightly less effective for this dataset. Overall, the careful design and optimization of the Custom LSTM model contributed significantly to its superior performance in phishing detection

Table 1: Performance Matrices of all Models on Website dataset

	Model Test	Accuracy	F1 Score	Test Loss
0	LSTM	0.918198	0.916852	0.217107
1	GRU	0.909886	0.908036	0.232197
2	Bidirectional LSTM	0.917760	0.916071	0.224658
3	Custom LSTM	0.926509	0.925267	0.203259

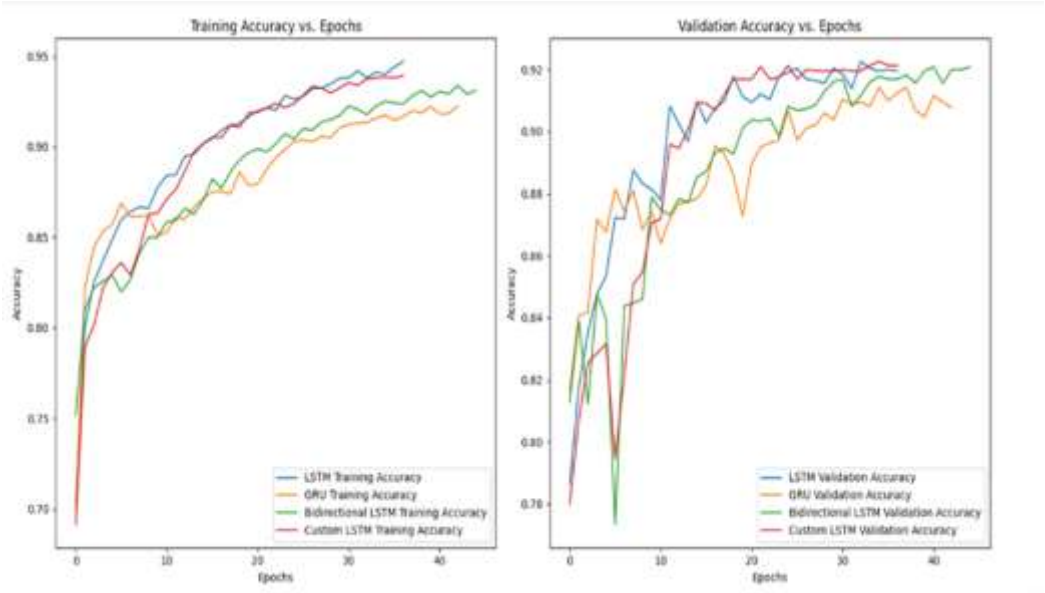

Figure 3: Graph of training and testing Accuracies vs Epoch for all models

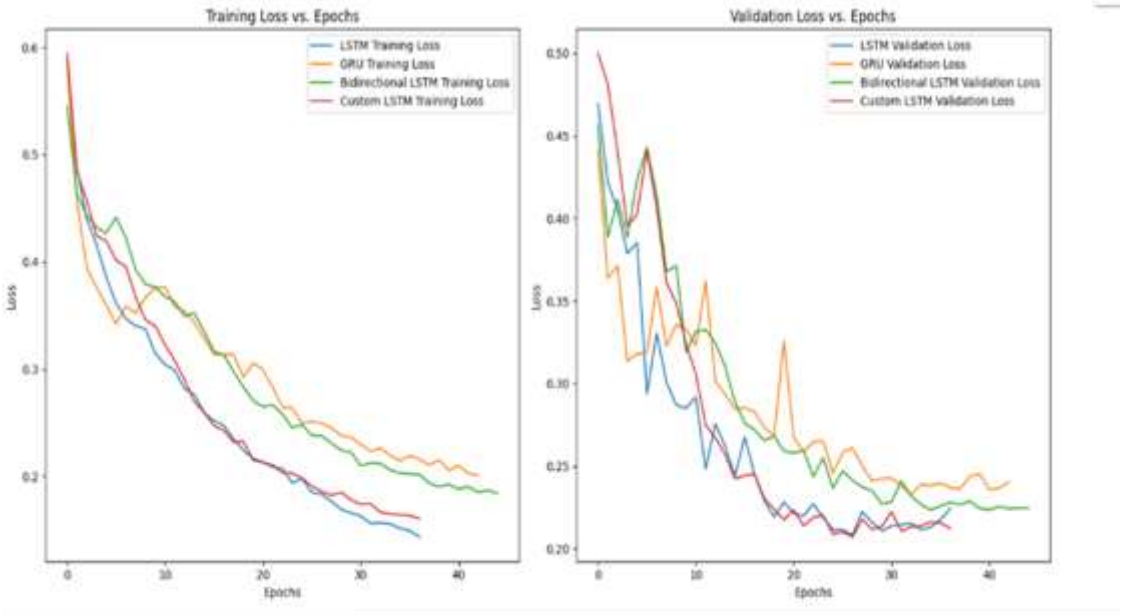

Figure 4: Graph of training and testing Loss vs Epoch for all models

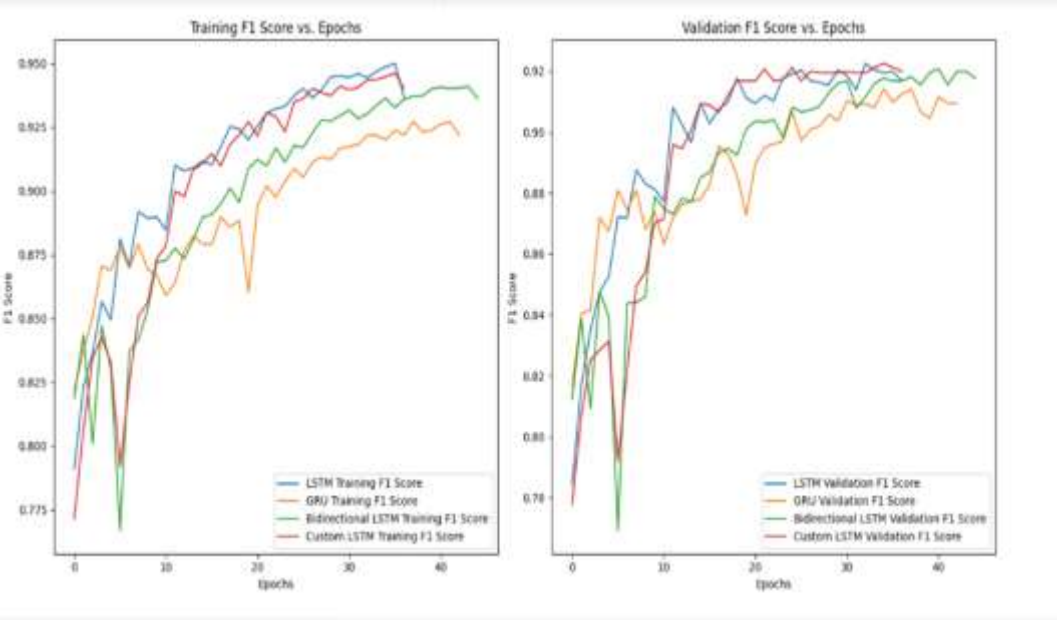

Figure 5: Graph of training and testing F1 Score vs Epoch for all models

Discussion of Findings

The focal point of the study was directed towards the advancement and assessment of a Custom Recurrent Neural Network (CRNN) model that was intricately customized to effectively identify fraudulent websites aimed at deceiving individuals, with the ultimate goal of improving the precision and effectiveness of existing methods for identifying such deceptive online platforms. Through the utilization of sophisticated machine learning strategies, specifically concentrated within the domain of neural networks, the CRNN model was meticulously crafted to scrutinize complex patterns and subtle attributes inherent in both URLs and the content of webpages. The CRNN model demonstrated promising results in detecting phishing websites, showcasing its ability to analyze sequential data effectively and differentiate between legitimate and malicious URLs. The model's performance was evaluated not only based on quantitative metrics but also on qualitative insights into its robustness and versatility. The findings indicated that the CRNN model could significantly enhance the accuracy and efficiency of phishing detection, potentially mitigating financial losses and reputational damage for organizations.

Conclusion

Summary of Findings

The findings of this project include the realization of the high rate of phishing attacks, targeted towards financial institutes.

Furthermore, the focal point of the study was directed towards the advancement and assessment of a Custom Recurrent Neural Network (CRNN) model that was intricately customized to effectively identify fraudulent websites aimed at deceiving individuals, with the ultimate goal of improving the precision and effectiveness of existing methods for identifying such deceptive online platforms. Through the utilization of sophisticated machine learning strategies, specifically concentrated within the domain of neural networks, the CRNN model was meticulously crafted to scrutinize complex patterns and subtle attributes inherent in both URLs and the content of webpages.

The custom LSTM model is composed of not just one, but two LSTM layers that are meticulously designed to effectively address the issue of overfitting by incorporating dropout layers. The first LSTM layer is equipped with a total of 128 units and is configured to return sequences, thereby enhancing the model's ability to comprehend the intricate patterns within the input data. On the other hand, the subsequent LSTM layer comprises 64 units but does not return sequences, offering a different perspective on the data analysis process.

The intricate design of this architecture plays a pivotal role in enabling the model to adeptly capture both short-term nuances and long-term dependencies present within the input sequences, fostering a comprehensive understanding of the underlying patterns.

In order to further enhance the robustness and generalization capabilities of the model, dropout layers are introduced with a dropout rate set at 0.2 subsequent to each LSTM layer. This strategic implementation serves the purpose of regularizing the model and safeguarding against overfitting by selectively deactivating certain units during the training phase. As a result of the model is incentivized to cultivate more resilient and generalized representations of the input data, thereby improving its overall performance

Moreover, to facilitate efficient feature extraction and classification, two dense layers are seamlessly integrated after the LSTM layers. The initial dense layer boasts a configuration of 32 units coupled with ReLU activation, a critical element that introduces non-linearity to the model and empowers it to discern intricate and complex patterns embedded within the data structure.

Ultimately, the culminating dense layer embodies a singular unit with sigmoid activation, operating as the crux of the binary classification process by making insightful predictions regarding the legitimacy or potentially malicious nature of a given website. The model's exceptional training accuracy of 95% and test accuracy of 92% further underscore its proficiency in navigating through the intricacies of the data.

Limitations

The limitations of this study are varied and encompass a wide range of challenges, starting with the difficulty in obtaining a comprehensive dataset that is rich in detailed features. This is further compounded by the challenge of effectively filtering and sorting through features that have low p ratings, which is essential for ensuring the accuracy and reliability of the model during the training process. It is imperative to address these issues meticulously as they directly impact the ability of the model to achieve a high level of accuracy in its predictions.

Conclusion

Based on recent reports from the Anti Phishing Working Group in 2023, there was a notable spike in phishing events as compared to other quarters in the 3rd quarter of 2023, and these attacks were targeted at the financial sector, causing a notable increase in identity theft which led to individuals losing valuable assets and personal information to unauthorized parties.

Experimental results indicated that the Default LSTM Model performed better than other variations of Recurrent Neural Network Models. Furthermore, A Custom Model was developed from the Default LSTM Model to further increase the accuracy and precision of the Model in tackling modern and recent phishing techniques.

The highest Accuracy Score, F1 Score and Test Loss for the LSTM Model was record at 91.8%, 91.6% and 21.7% respectively. This was the highest on record for the RNN variation Models, subsequently a Custom Model was derived from the default LSTM Model which was trained and tested with the dataset.

Hence, the Evaluation Metric obtained from the Custom LSTM Model are as follows, Accuracy Score, F1 Score and Test Loss are 92.6%, 92.5% and 20.3% respectively.

Recommendation

To tackle the developing risks of phishing, which will likely lead to data loss and identity theft, companies, government parastatals and financial institutes are advised to allocate resources towards strong email screening systems, anti-fraud tools, staff education initiatives, multi-layer authentication, website link inspection procedures, and preparedness strategies for unexpected event

Contribution to Knowledge

The research suggests a custom multimodal strategy for identifying harmful URLs by employing various modified recurrent neural network architectures, demonstrating the efficacy of LSTM, Bi-LSTM, and GRU within this domain.

The research contributes to the advancement of cybersecurity measures by demonstrating the efficacy of deep learning techniques in combating phishing attacks and protecting users from online threats.

References:

Research Papers:

1. APWG. (2023). Anti-Phishing Working Group. Barcelona: APWG. Retrieved from https://docs.apwg.org/reports/apwg_trends_report_q1_2023.pdf
2. Balogun, A., Adewole, K., Raheem, M., Akande, O., Usman-Hamza, F., Mabayoje, M., & Akintola, A. e. (2021). Improving the phishing website detection using empirical analysis of Function. Heliyon, 7.
3. Jones, A. (2020). The Evolution of Online Fraud: A Comprehensive Review. Journal of Cybersecurity Studies, 45-62.
4. Li, T., Kou, G., & Peng, Y. (2020). Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation. Inf. Syst, 91.
5. Mehmet Korkmaz, O. K. (2020). Detection of phishing websites by using machine learning-based URL analysis. 11nth International Conference on Computing, Communication and Networking Technologies (ICCCNT).
6. Minocha, S., & Singh, B. (2022). A novel phishing detection system using binary modified equilibrium optimizer for feature selection. Comput. Electr. Eng, 98.
7. Morgan, Steve. (2022, Oct 17). Ventures, Cybersecurity. Retrieved from Cyber Security Ventures: https://cybersecurityventures.com/cybercrime-to-cost-the-world-8-trillion-annually-in-2023/
8. Sunil Chaudhary. (2012). Recognition of phishing attacks utilizing anomalies in phishing websites. University of Tempere.

9. Verizon. (2021). Data Breach Investigations Report. Retrieved from Data Breach Investigations Report: https://enterprise.verizon.com/resources/reports/dbir/.