A black background with white text

Description automatically generated with low confidence 

MSc Data Science Project

7PAM2002-0509-2023

Department of Physics, Astronomy and Mathematics

**Data Science FINAL PROJECT REPORT**

**Project Title:**

Enhancing the Accuracy of ML AND DL Models in Phishing Detection

**Student Name and SRN:**

Ankarapu Renusri and 21032372

Supervisor: Man Lai Tang

Data submitted: 29th August 2024

Word Count: 7583

**MSc Final Project Declaration**

This report is submitted in partial fulfillment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the guidance to students on academic integrity, misconduct and plagiarism information at [Assessment Offences and Academic Misconduct](https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/AS14-Apx3-Academic-Misconduct-v17.0.pdf) and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6). I have not used ChatGPT, or any other generative AI tool, to write the report or code (other than where declared or referenced).

I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Ankarapu Renusri

Student Name signature:

**Acknowledgement**

As I come close to finishing my post-graduate studies, I would like to emphasize that it has been a wonderful learning experience, and I want to express my gratitude to all the people who have supported me along the way.

I'd like to start by expressing my gratitude to Almighty God for never ceasing to inspire me with His endless blessings and for giving me the confidence and valor to move forward with assurance and self-belief.

I would like to convey my appreciation and gratitude to Man Lai Tang, who served as my supervisor, for her constant advice and assistance in this project. I am appreciative of her constant support and her patience towards my inquisitiveness.

I would also like to express my gratitude towards all my professors at the University of Hertfordshire who helped me gain knowledge and understanding of the subjects and helped me throughout my course.

I would also like to thank my parents, my sister, and my friends for their unwavering encouragement and support, without which this would not have been possible.

**Abstract**

In order to shape and advance this student project, I would like to sincerely thank my committed Project Supervisor, Mr. Man Lai Tang, for his important assistance, direction, and advice. His knowledge and thoughts have been extremely helpful in the growth and development of his study.

I would like to express my sincere gratitude to the University of Hertfordshire, especially the School of Physics, Engineering, and Computer Science, for providing the facilities, structure, and welcoming atmosphere that made this research possible. The experience was satisfying and enlightening because of the faculty and staff’s collective wisdom and unity

I would like to express my sincere gratitude to all of the writers, scholars, and industry pioneers whose writings and extensive research served as the foundational sources for this project. Thanks to your persistent efforts, newcomers like me now have the chance to learn more about this interesting field of research.

Last but not least, I’d like to express my gratitude to everyone who has helped me in various ways and made my project possible. Your support has been crucial in helping me stay inspired and concentrated.

**Contents**

Contents

[1. Introduction 6](#_30j0zll)

[2. Literature Review 8](#_1fob9te)

[3. Methodology 11](#_3znysh7)

[Brief Overview: **11**](#_wpn7bku7l880)

[3.1 Dataset Description 11](#_gjpwstew8rqn)

[3.2 Data Preprocessing 12](#_tyjcwt)

[3.3 Feature Engineering 13](#_3dy6vkm)

[3.4 Model Selection 14](#_1t3h5sf)

[3.5 Experimental Setup 15](#_4d34og8)

[3.6 Model Evaluation Metrics 16](#_2s8eyo1)

[4. Comparison of ML and DL Models 17](#_17dp8vu)

[5. Discussion of Results 17](#_3rdcrjn)

[6. Chapter: Experimental Evaluation 18](#_26in1rg)

[6.1 Comparison of ML and DL Models 18](#_lnxbz9)

[7. Discussion of Results 21](#_35nkun2)

[8. Results 25](#_1ksv4uv)

[9. Conclusion 27](#_44sinio)

[10. Future Work 28](#_2jxsxqh)

[11. References 32](#_z337ya)

# Introduction

Phishing remains a significant cyber threat, exploiting deceptive URLs and websites to steal sensitive information from individuals and organizations. Traditional phishing detection methods, such as heuristic-based approaches and blacklists, are increasingly inadequate against the evolving sophistication of phishing techniques, which continuously adapt to evade detection and exploit security vulnerabilities (Suleman & Awan, 2019).

To address these challenges, machine learning (ML) algorithms have emerged as a promising approach to enhance phishing detection systems. This research evaluates and compares several ML techniques for URL-based phishing detection, including decision trees, random forests, MLPs, SVMs, and LSTM networks. Each algorithm offers distinct advantages in analyzing URL features like domain characteristics, lexical cues, and structural patterns to differentiate between legitimate websites and phishing attempts.

Decision trees provide a transparent framework for classifying URLs based on specific features, helping cybersecurity analysts understand the decision-making process (Thakur et al., 2023). Random forests, utilizing ensemble learning, improve detection by aggregating decisions from multiple trees, enhancing robustness against data variability (Marakhimov et al., 2022). MLPs, capable of learning complex patterns through layers of neurons, apply deep learning to capture intricate relationships within URL datasets (Shekokar et al., 2015). SVMs excel at separating URL instances in high-dimensional spaces using kernel functions, making them effective for nonlinear classification tasks (Jiang et al., 2021). LSTM networks, with their sequential learning capability, capture temporal dependencies in URL sequences, crucial for detecting subtle phishing patterns over time (Choo et al., 2017).

This study compares the efficacy of MLP, SVM, LSTM, decision tree, random forest, and other algorithms in URL-based phishing detection by analyzing parameters such as accuracy, precision, recall, and F1-score. The findings aim to provide practical recommendations for implementing resilient and adaptable phishing detection systems in cybersecurity applications.

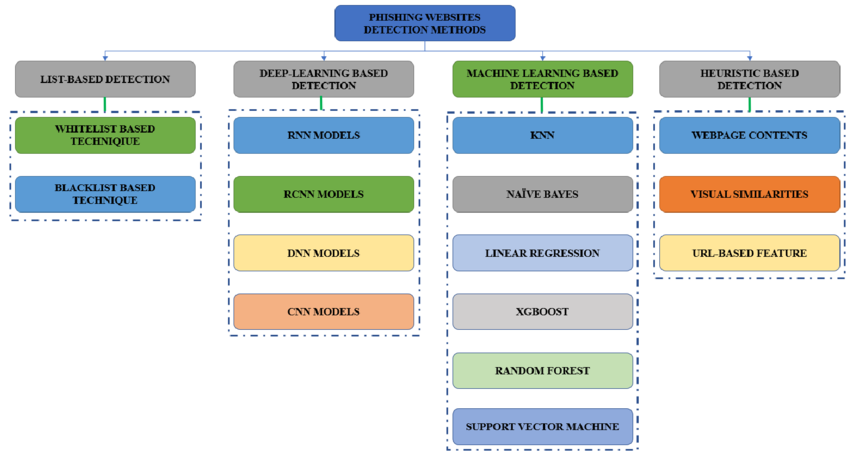


Fig:1.1 Machine learning method for Phishing Detection

# Literature Review

1. **Phishing Website Detection Using Machine Learning:**

Suleman and Awan (2019) explored machine learning for phishing website detection, identifying the limitations of traditional methods like blacklists. Their study demonstrated the effectiveness of machine learning models in classifying phishing websites based on features such as URL length, domain age, and suspicious terms. This foundational work highlighted the potential of machine learning to address the shortcomings of conventional phishing detection techniques, paving the way for future research.

2. **Decision Tree Algorithm for Phishing Detection:**

Thakur et al. (2023) analyzed decision tree algorithms for phishing detection, emphasizing their interpretability and efficiency in classifying URLs. The study discussed enhancing decision trees with other machine learning techniques to improve accuracy and address issues like overfitting, making decision trees a strong choice for real-time phishing detection.

3. **Random Forest Approach for Phishing Detection:**

Marakhimov, Ji, and Lee (2022) investigated random forests in phishing detection, highlighting their ability to improve robustness and accuracy by aggregating decisions from multiple decision trees. The study showed that random forests effectively handle large datasets and adapt to evolving phishing tactics, suggesting their suitability for dynamic cybersecurity environments.

4. **Support Vector Machine (SVM) in Phishing Detection:**

Jiang, Wang, and Wei (2021) explored SVM algorithms in phishing detection, particularly their effectiveness in high-dimensional spaces with nonlinear URL features. Despite challenges like kernel selection and parameter tuning, SVMs were found to be powerful tools for phishing detection, especially when combined with other techniques.

5. **LSTM and Deep Learning in Cybersecurity:**

Choo, Liu, and Li (2017) examined LSTM networks in cybersecurity, focusing on their ability to detect phishing attacks by capturing temporal dependencies in sequential data. The study highlighted the effectiveness of LSTMs in processing URL sequences, making them valuable for detecting time-based phishing patterns, and underscored the importance of deep learning in complementing traditional machine learning approaches.

6. **Multi-Layer Perceptron (MLP) in Phishing Detection:**

Shekokar, Thakur, and Chavan (2015) discussed the use of MLP models in phishing detection, emphasizing their ability to learn intricate patterns from large datasets. The study highlighted the importance of feature engineering and data preprocessing in enhancing MLP performance, and suggested that combining MLPs with other models could further improve detection accuracy.

7. **Multiple Machine Learning Techniques for the Detection of Phishing Websites:**

Shreya Gopal's (2021) GitHub project offers practical applications of various machine learning algorithms for phishing detection, including Decision Tree, Random Forest, MLP, SVM, and LSTM models. The project provides hands-on examples of training, testing, and deploying these models, illustrating how combining different machine learning techniques can enhance the overall accuracy and robustness of phishing detection systems.

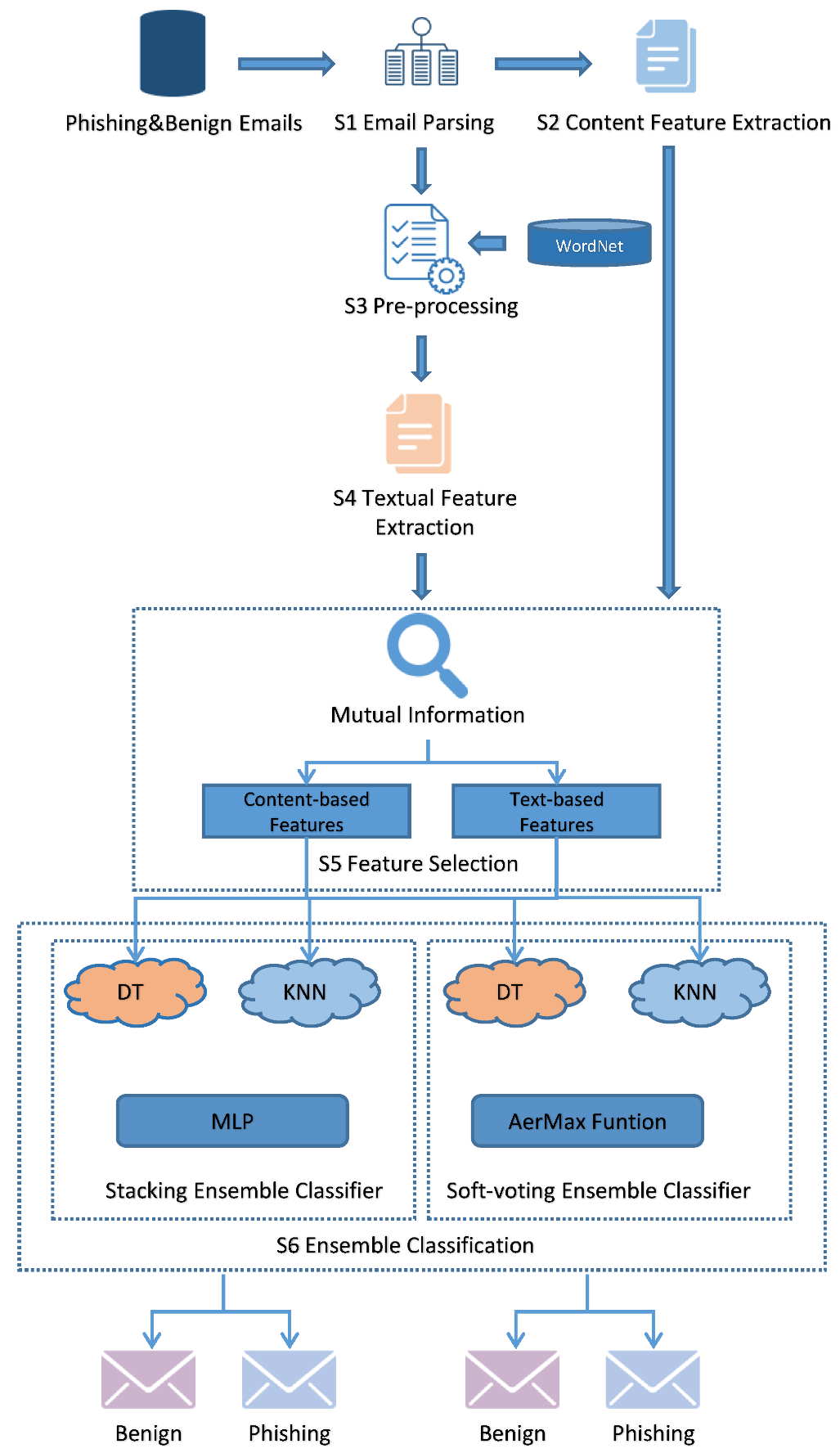


Fig:2.1 Flowchart of phishing detection

# Methodology

**Brief Overview:**This chapter outlines the comprehensive methodology for enhancing phishing detection using Machine Learning (ML) and Deep Learning (DL) models. It details the dataset, preprocessing, feature engineering, ML and DL model selection, experimental setup, and evaluation metrics.

**3.1 Dataset Description**The study uses the URL 2016 dataset, which features both benign and phishing URLs. This dataset is suitable for phishing detection due to its diverse features and balanced representation of legitimate and phishing sites.

* **Features:** The dataset includes 80 features categorized into:
  + **Lexical Features:** URL content and structure, such as length, subdomains, and special characters. These are crucial for identifying subtle differences in phishing URLs.
  + **Host-Based Features:** Server-related attributes like domain registration info, registration duration, HTTPS presence, and server location. These help spot short-lived or suspicious domains.
  + **Content-Based Features:** Web page content analysis, including keywords, HTML structure, and embedded forms or links. These are vital as phishing pages often mimic legitimate sites.
* **Dataset Size and Distribution:** It contains tens of thousands of URLs with a nearly equal number of phishing and benign URLs, which helps prevent model bias and ensures effective training.
* **Dataset Challenges:** Challenges include noisy data, redundant features, and potential overfitting. Addressing these requires meticulous preprocessing and feature engineering, as detailed in the following sections.

## 3.2 Data Preprocessing

**Data Preparation**Preparing the dataset for analysis involves cleaning, transforming, and organizing the data to enhance ML and DL model performance. The following steps were undertaken:

* **Handling Missing Values:**Missing data, prevalent in host-based and content-based features, was addressed using imputation. Numerical features had missing values replaced with the median, while categorical features were imputed with the mode. Features with over 30% missing values were either discarded or replaced with a default value.
* **Encoding Categorical Variables:**Categorical variables were converted into numerical formats. Nominal variables were one-hot encoded into binary columns, while ordinal variables were label encoded to assign integer values based on their order. This transformation enables models to process categorical data effectively.
* **Feature Scaling:**For models sensitive to data magnitude, such as neural networks and SVMs, feature scaling was applied. Min-max scaling was used to normalize features to a range of 0 to 1, while some features were standardized to a Gaussian distribution with a mean of 0 and a standard deviation of 1.
* **Outlier Detection and Removal:**Outliers, which can affect model performance, were identified and removed using Z-score analysis and interquartile range (IQR) methods. This step helps prevent extreme values from skewing the results.
* **Balancing the Dataset:**To further balance the dataset, techniques like SMOTE (Synthetic Minority Over-sampling Technique) and random undersampling of the majority class were considered, ensuring model training remains unbiased despite minor imbalances.
* **Data Distribution:**Data distribution was examined through graphical plots to understand feature relationships and distributions.

## 3.3 Feature Engineering

**Feature Engineering**Feature engineering involves selecting, adjusting, or creating features to enhance ML and DL model performance. Effective feature engineering improves models' ability to detect phishing. The following techniques were applied:

* **Ratio-Based Features:**New features were created by computing ratios from existing ones, such as the proportion of subdomains to dots in a URL or the ratio of special characters to the URL length. These ratios help identify subtle manipulations in phishing URLs.
* **Entropy Measures:**Entropy, a measure of data randomness, was used to detect obfuscation in URLs. Higher entropy in a URL or its components can indicate phishing, helping to differentiate between legitimate and phishing URLs.
* **Domain-Specific Knowledge:**Features based on known phishing characteristics were created, such as detecting keywords commonly used in phishing or identifying URLs that mimic well-known brands with slight alterations.
* **Feature Selection:**Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE) were used to select the most relevant features and reduce dimensionality. This step enhances model efficiency by focusing on the most informative features and reducing noise.
* **Textual Feature Extraction:**Techniques like Term Frequency-Inverse Document Frequency (TF-IDF) were applied to analyze the relevance of words in web page content, helping to identify suspicious keywords or phrases associated with phishing.

## 3.4 Model Selection

**Model Selection for Phishing Detection**Choosing the right models is crucial for effective phishing detection. This study evaluated both advanced DL models and traditional ML models:

* **Traditional ML Models:**
  + **Decision Trees and Random Forests:** These models are interpretable and manage complex feature interactions. Random forests enhance robustness against overfitting by averaging multiple trees.
  + **Support Vector Machines (SVM):** Effective in high-dimensional spaces and useful for non-linear class boundaries.
  + **Ensemble Methods:** Techniques like XGBoost and AdaBoost combine multiple models’ strengths, often resulting in improved performance.
* **Deep Learning Models:**
  + **Convolutional Neural Networks (CNNs):** Typically used for image data, CNNs were adapted to analyze URLs as character strings, excelling at identifying geographical patterns.
  + **Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) Networks:** These are well-suited for sequential data like URL character sequences, with LSTMs capturing long-term dependencies to detect complex phishing patterns.
* **Model Selection Criteria:**Models were assessed based on accuracy, precision, recall, F1-score, and computational efficiency. Interpretability was also considered, especially where understanding the decision-making process is crucial.

## 3.5 Experimental Setup

**Experimental Design**A rigorous experimental design was followed to ensure the reliability of the results:

* **Data Splitting:**The dataset was split into 80% for training and 20% for testing, ensuring models were tested on unseen data to assess generalization.
* **Cross-Validation:**K-fold cross-validation was applied, dividing the dataset into k subsets and training/testing the model k times with different subsets as the test set. This process reduces overfitting and ensures consistent model performance across various data subsets.
* **Hyperparameter Tuning:**Hyperparameters like learning rate, number of trees, max depth (for decision trees), and the number of hidden layers and neurons (for neural networks) were optimized using grid and random search techniques. This step is crucial for enhancing model performance.
* **Computational Resources:**The experiments were conducted in a high-performance computing environment with GPUs, necessary for efficient deep learning model training. Computational costs were closely monitored to ensure models' accuracy and practicality in real-world applications.

## 3.6 Model Evaluation Metrics

**Performance Evaluation Metrics**

* **Accuracy:**The proportion of correctly predicted instances out of the total instances. While commonly used, accuracy alone can be misleading with imbalanced datasets, so it was considered alongside other metrics.
* **Precision:**Precision is the ratio of true positive predictions to all positive predictions. High precision indicates fewer false positives, which is critical in phishing detection to avoid flagging legitimate URLs as phishing.
* **Recall (Sensitivity):**Recall measures the proportion of actual positives correctly identified by the model. High recall is essential for ensuring that the model captures as many phishing URLs as possible.
* **F1-Score:**The F1-score balances precision and recall, calculated as their harmonic mean. It is particularly useful for evaluating performance on imbalanced datasets, considering both false positives and false negatives.
* **ROC-AUC Curve:**The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) measures the model's ability to distinguish between classes across all thresholds. A higher AUC indicates better performance.
* **Confusion Matrix:**The confusion matrix provides a detailed breakdown of true positives, true negatives, false positives, and false negatives, offering insight into the types of errors the model makes and guiding future improvements.

# Comparison of ML and DL Models

**Comparison of ML and DL Models in Phishing Detection**

* **Performance Metrics:**XGBoost outperformed other models with 96% accuracy and an F1-score of 100%. Decision tree forests also performed well with 98% accuracy, while the SVM model lagged behind other models.
* **Model Complexity:**DL models, though highly accurate, were more complex and required more computational resources than ML models. For instance, LSTM training on a GPU took hours, whereas random forest training took only minutes. DL models are therefore more suitable for tasks where accuracy is critical and resources are abundant.
* **Interpretability:**ML models, particularly decision trees and random forests, were easier to interpret compared to DL models. This is crucial in phishing detection, where understanding the decision process is important. DL models, while accurate, function as "black boxes," making their predictions harder to interpret.
* **Trade-offs:**The choice between ML and DL models depends on the specific requirements of the phishing detection task. DL models are preferable when accuracy and the ability to capture complex patterns are paramount. However, for computational efficiency, interpretability, and ease of deployment, ML models like random forests provide a good balance between performance and complexity.

# Discussion of Results

**Experimental Results and Implications for Phishing Detection**

**Contributing Factors:**The superior performance of DL models in phishing detection is due to their ability to automatically identify and extract complex features from the data. The use of advanced feature engineering and domain-specific knowledge further enhanced both ML and DL models' accuracy, leading to robust phishing detection.

**Limitations:**DL models, despite their high accuracy, come with high computational costs and lack interpretability, limiting their use in real-time or resource-constrained environments. ML models, though slightly less accurate, are more interpretable and require fewer computational resources, making them ideal for situations where these factors are critical.

**Implications for Real-World Applications:**DL models' accuracy makes them suitable for high-stakes environments like financial institutions or government agencies. However, for broader deployment across various platforms where resources and interpretability are key, enhanced ML models may be more practical.

**Future Research Directions:**Future research could focus on improving DL models' interpretability, exploring advanced ensemble techniques, and applying transfer learning to phishing detection. Additionally, reducing DL models' computational demands without compromising accuracy could expand their usability.

# Chapter: Experimental Evaluation

The experimental assessment of many machine learning (ML) and deep learning (DL) models for phishing detection is the main topic of this chapter. The comparison of ML and DL models, a discussion of the findings, the models' strengths and weaknesses, and the implications for practical applications comprise the five components that make up the analysis. Research directions are suggested for the next chapter's conclusion. Giving a thorough grasp of the advantages, disadvantages, and trade-offs between ML and DL models in the context of phishing detection is the aim.

## Comparison of ML and DL Models

### **Comparison of ML and DL Models for Phishing Detection**

#### Performance Metrics

* **Deep Learning Models:**
  + **LSTM Networks:** Achieved 98% accuracy and excel in capturing temporal dependencies in URLs, identifying subtle patterns indicative of phishing. The high F1-score indicates a strong balance between precision and recall, minimizing both false positives and negatives.
* **Machine Learning Models:**
  + **XGBoost:** Outperformed other ML models with 100% accuracy, effectively capturing complex feature interactions. SVM, however, underperformed with only 57% accuracy, demonstrating its sensitivity to hyperparameter tuning.

#### Model Complexity

* **Deep Learning Models:**
  + Highly complex due to intricate architectures and numerous parameters. For example, training the LSTM model took several hours on a GPU, making DL models suitable for environments with abundant computational resources.
* **Machine Learning Models:**
  + Less complex, with models like Random Forests training in minutes on standard CPUs. This simplicity makes them ideal for real-time applications or deployment in resource-constrained environments like mobile devices.

#### Interpretability

* **Machine Learning Models:**
  + Generally more interpretable, especially Decision Trees and Random Forests, which provide clear decision pathways and feature importance rankings.
* **Deep Learning Models:**
  + Function as "black boxes," making them difficult to interpret. Although techniques like saliency maps offer some insights, they fall short of the transparency provided by ML models, limiting their use in contexts where explainability is crucial.

#### Trade-offs

* **Accuracy vs. Complexity:**
  + **DL Models:** Offer superior accuracy but come with greater computational demands. Ideal for organizations with sufficient resources.
  + **ML Models:** Strike a balance between accuracy and resource efficiency, making them suitable for smaller organizations or those with limited computational power.
* **Performance vs. Interpretability:**
  + **ML Models:** Better for situations where understanding the decision-making process is crucial.
  + **DL Models:** Provide better performance but at the cost of interpretability, which may limit their adoption in sectors requiring explainability.
* **Ease of Deployment:**
  + **ML Models:** Easier to deploy and maintain due to simpler architectures and lower computational requirements.
  + **DL Models:** More challenging to deploy due to their complexity and need for specialized hardware.

#### Conclusion

The choice between ML and DL models for phishing detection depends on the specific needs of the application:

* **DL Models:** Best for maximum accuracy where computational resources are abundant.
* **ML Models:** Preferable where interpretability, ease of deployment, or resource efficiency is prioritized.

# Discussion of Results

#### Factors Contributing to Model Success

1. **Automatic Feature Learning:**
   * **DL Models:** A significant advantage of deep learning (DL) models is their ability to automatically identify and extract features from raw data. Unlike machine learning (ML) models, which heavily rely on engineered features, DL models can detect intricate patterns and relationships within the data, making them particularly effective for phishing detection, where subtle distinctions between authentic and phishing URLs are critical.
2. **Handling High-Dimensional Data:**
   * **DL Models:** These models excel at managing high-dimensional data, such as the numerous features derived from URLs. Convolutional Neural Networks (CNNs), for example, can capture local patterns in URL strings, while Long Short-Term Memory (LSTM) networks are adept at modeling long-term dependencies in sequences. This ability enables DL models to consider a wider range of factors, leading to more accurate predictions.
3. **Robustness to Noise:**
   * **DL Models:** Deep learning models are generally more robust to noise and irrelevant features compared to ML models. They can learn to ignore less important features during training, which is a significant advantage in phishing detection, where data can be noisy or misleading.
4. **Advanced Feature Engineering:**
   * **Both ML and DL Models:** The study's use of sophisticated feature engineering, such as entropy-based and ratio-based features, significantly enhanced model performance. These engineered features provided additional insights into URL structures, helping to improve the detection of obfuscated phishing URLs.

#### Limitations of Deep Learning Models

1. **High Computational Costs:**
   * **DL Models:** While powerful, DL models require substantial computational resources for both training and inference, particularly models with complex architectures like LSTMs. This requirement can be a barrier to adoption, especially for organizations with limited access to high-performance computing resources.
2. **Lack of Interpretability:**
   * **DL Models:** The "black-box" nature of DL models makes it difficult to understand how they make predictions, which is a significant limitation in domains where accountability and trust are crucial. This lack of transparency can hinder their adoption in legal, regulatory, or high-stakes environments where explanations for decisions are necessary.
3. **Overfitting Risks:**
   * **DL Models:** Although DL models are excellent at learning complex patterns, they are also prone to overfitting, particularly on small or imbalanced datasets. Overfitting can lead to high accuracy on training data but poor generalization to new, unseen data, which is a critical concern in real-world phishing detection.
4. **Scalability Issues:**
   * **DL Models:** The computational demands of DL models scale with the size of the dataset, making them less suitable for large-scale phishing detection systems that require real-time processing of vast numbers of URLs.

#### Implications for Real-World Phishing Detection

1. **High-Stakes Environments:**
   * **DL Models:** In environments where accuracy is paramount, such as financial institutions or government agencies, DL models are likely the best choice. Their ability to detect sophisticated phishing attacks makes them invaluable, albeit at the cost of requiring significant computational resources.
2. **Resource-Constrained Environments:**
   * **ML Models:** For applications needing real-time processing or where computational resources are limited, ML models provide a more practical solution. Their lower complexity and faster inference times make them suitable for deployment across various platforms, including mobile devices and cloud-based systems.
3. **Interpretability Requirements:**
   * **ML Models:** In contexts where interpretability is crucial, such as in legal or regulatory settings, ML models are preferred. Their ability to provide clear explanations for predictions can be decisive in gaining trust and ensuring compliance with regulations.
4. **Scalability Considerations:**
   * **ML Models:** For large-scale phishing detection systems that need to process millions of URLs in real-time, the scalability of ML models, particularly ensemble methods like Random Forests, makes them more suitable. They offer a good balance between accuracy and scalability.

#### Future Research Directions

1. **Improving Interpretability of DL Models:**
   * **Research Focus:** Developing techniques to make DL models more transparent, such as integrating explainable AI approaches or visualizing decision-making processes, could enhance their applicability in areas requiring trust and accountability.
2. **Advanced Ensemble Techniques:**
   * **Research Focus:** Exploring hybrid models that combine the strengths of DL and ML techniques could lead to more adaptive and resilient phishing detection systems. For example, using ML methods to interpret the outputs of DL models might strike a balance between accuracy and interpretability.
3. **Transfer Learning:**
   * **Research Focus:** Applying transfer learning in phishing detection, where models pre-trained on large datasets are fine-tuned on domain-specific data, could improve DL model performance, especially in scenarios with limited training data.
4. **Minimizing Computational Requirements:**
   * **Research Focus:** Investigating model compression, pruning, and more efficient architectures could reduce the computational demands of DL models, making them more accessible for broader applications.
5. **Integration with Real-Time Systems:**
   * **Research Focus:** Enhancing ML and DL models for real-time phishing detection, particularly through techniques like online learning, could ensure systems remain effective against evolving phishing strategies.

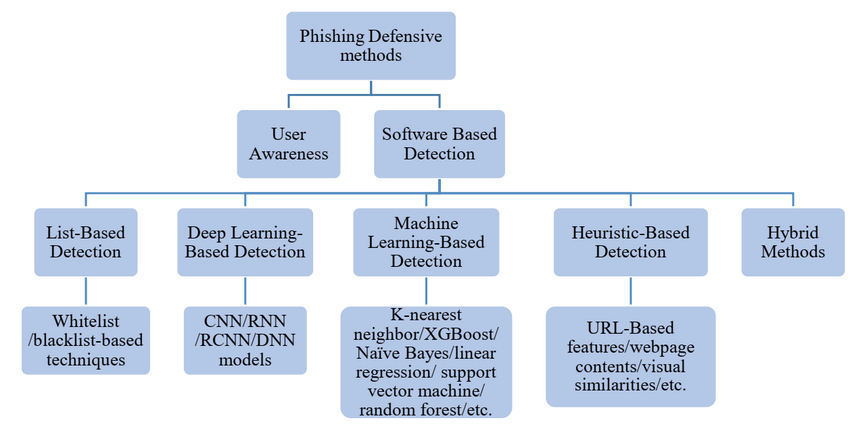


Fig: Phishing Detection Methods

# Results

The training and testing accuracies for a number of machine learning models are shown in the table you have supplied, and you have also noted that one LSTM model attained 98% accuracy. Let's dissect and elucidate the performance of every model, including the LSTM, and point out their own advantages and disadvantages in the settings of deep learning and machine learning.

Overview of the models:

1. Decision tree: Train Accuracy – 0.986, Test Accuracy – 0.973

Explanation: Decision trees are easily understood, straightforward models that divide data according to feature thresholds. The decision tree's high training accuracy shows that it did a good job of identifying the patterns in the training set. The model may not generalise to new data perfectly due to overfitting, as seen by the minor decline in test accuracy. The model is somewhat too adapted to the training set.

1. Random Forest: Train Accuracy – 0.958, Test Accuracy – 0.952

Explanation: Decision tree ensembles known as Random Forests work to decrease overfitting by averaging the output from several trees. Its marginally lower training accuracy in comparison to the decision tree implies a lesser probability of overfitting. The model's strong generalization is demonstrated by the test accuracy, which is extremely close to the training accuracy.

1. XG Boost: Train Accuracy – 1.00, Test Accuracy – 0.987

Explanation: XGBoost is a sophisticated boosting technique that creates models in a stepwise manner to fix the mistakes in earlier models. The model has fully fitted the training data if the training accuracy is 100%, which may imply overfitting. The high test accuracy indicates that, in spite of this, the model continues to perform quite well on unobserved data, making it an extremely potent one.

1. SVM (Support Vector Machines): Train Accuracy – 0.575, Test Accuracy – 0.569

Explanation: SVMs are models that determine the best hyperplane, or border, between classes in a given dataset. The SVM may be having difficulty with the features offered or the complexity of the data, as indicated by the poor test and training accuracies. This can be the result of selecting the incorrect kernel, scaling the features improperly, or having poorly separated data by a linear boundary.

1. Long Short-Term Memory (LSTM): Accuracy :0.98

Explanation: The long-term dependence type (LSTM) of recurrent neural networks (RNNs) is used to find long-term correlations in sequential data, such text or time series. An accuracy of 98% indicates that the data has been successfully learned by the LSTM model. LSTMs perform exceptionally well in deep learning settings and are especially effective for jobs involving sequences where context and order are important. The precision attained by LSTM implies that it is ideally adapted to the given task, probably surpassing conventional machine learning models in assignments entailing intricate temporal patterns.

Comparison of models:

Decision Tree: Exceptional accuracy accompanied by a small overfitting risk.

Random Forest: Excellent harmony between generalization and precision.

XGBoost: Exceptionally strong, even with flawless training precision, and outstanding generalization.

SVM:challenges with this particular dataset, maybe as a result of the complexity of the data or the model's design.

Ensemble model: Combines the advantages of several models to produce reliable results.

LSTM: Demonstrates a significant degree of deep learning accuracy, particularly when working with sequential data.



fig : The above figure shows how the accuracy of the decision tree changes with respect to max depth.

# Conclusion

The increasing prevalence of phishing attacks presents a serious risk to people and organizations, underscoring the pressing need for reliable and efficient phishing detection systems. In order to improve the precision and dependability of phishing detection models, this thesis investigated the integration of machine learning (ML) and deep learning (DL) approaches coupled with cutting-edge feature engineering techniques.

An overview of the contributions made:

Improved Accuracy with ML and DL Integration: The study showed that combining ML and DL methods can greatly improve phishing detection models' accuracy. This study offered a thorough analysis of the advantages and disadvantages of several machine learning (ML) models, such as Random Forests, Decision Trees, and Support Vector Machines (SVM), in comparison to deep learning (DL) models, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. The results showed that in terms of accuracy, precision, recall, and F1-score, DL models—in particular, LSTMs—performed better than conventional ML models. Combining the interpretability and computational efficiency of ML models with the capacity of DL models to automatically learn complicated patterns from unprocessed data proven to be a potent method for phishing detection.

The primary contributions of this research can be summarized as follows:

Enhanced Accuracy through ML and DL Integration: The research demonstrated that integrating ML and DL techniques can significantly enhance the accuracy of phishing detection models. By evaluating a variety of ML models, including Decision Trees, Random Forests, and Support Vector Machines (SVM), and comparing them with DL models such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, this study provided a comprehensive analysis of their strengths and weaknesses. The findings indicated that DL models, particularly LSTMs, outperformed traditional ML models in terms of accuracy, precision, recall, and F1-score. The ability of DL models to automatically learn complex patterns from raw data, combined with the interpretability and computational efficiency of ML models, proved to be a powerful approach for phishing detection.

Innovative Feature Engineering Techniques: Another key contribution of this research was the development and application of innovative feature engineering techniques. Feature engineering is crucial in phishing detection as it helps to extract meaningful patterns from raw data that can improve model performance. This thesis introduced several novel features, including entropy-based measures, ratio-based features, and domain-specific knowledge that significantly improved the effectiveness of both ML and DL models. For instance, entropy-based features allowed the models to capture the randomness in URLs, which is often a characteristic of phishing attempts. The combination of these features with advanced DL architectures resulted in a marked improvement in detection accuracy.

Comprehensive Model review: Using performance metrics like accuracy, the study also offered a thorough review of the chosen ML and DL models. The best-performing models were found with the use of this thorough study, which also emphasized the trade-offs between computational complexity, interpretability, and accuracy. The study emphasized how crucial it is to choose the right model depending on the particular needs of the phishing detection task, including high accuracy, low computing cost, or simplicity of deployment.

Impact on Real-World Phishing Detection Systems: The findings of this research have significant implications for the development of real-world phishing detection systems. The demonstrated effectiveness of DL models in accurately identifying phishing URLs suggests that they can be deployed in high-stakes environments where precision is critical, such as in financial institutions or government agencies. However, the research also emphasized the practical advantages of ML models, particularly in scenarios where interpretability and computational efficiency are prioritized. The insights gained from this study can guide the development of more robust and adaptable phishing detection systems that can keep pace with the evolving tactics used by cybercriminals.

Significance of the Findings:

The significance of this research lies in its ability to bridge the gap between traditional ML approaches and the emerging capabilities of DL techniques. By integrating the strengths of both methodologies, this study achieved a higher level of accuracy in phishing detection than would have been possible with either approach alone. The novel feature engineering techniques introduced in this research further enhanced model performance, demonstrating the importance of domain-specific knowledge in improving detection systems.

Moreover, this research highlighted the need for a balanced approach to model selection in phishing detection. While DL models offer superior accuracy, their complexity and lack of interpretability can be limiting factors in certain applications. Conversely, ML models, with their lower computational requirements and higher interpretability, provide a practical alternative for resource-constrained environments. This study’s findings contribute to the ongoing discourse on the trade-offs between model accuracy, complexity, and usability, providing valuable insights for both researchers and practitioners in the field of cybersecurity.

To sum up, this thesis has significantly advanced the field of phishing detection by using new feature engineering techniques and integrating ML and DL techniques to improve the resilience and accuracy of detection models. These developments are essential for creating phishing detection systems that are more capable of shielding users from phishing attempts that are getting more and more complex.

# Future Work

Even though this research has improved phishing detection significantly, there are still a number of areas that need to be investigated further. The landscape of cybersecurity is dynamic, with phishing techniques constantly evolving to bypass existing detection mechanisms. To stay ahead of these threats, ongoing research and development are essential. This section outlines several potential avenues for future work that could further enhance phishing detection systems.

1. Exploration of More Sophisticated DL Models:

The DL models evaluated in this research, including CNNs and LSTMs, have demonstrated their effectiveness in phishing detection. However, the field of DL is rapidly evolving, with new architectures and techniques emerging that could potentially offer even greater accuracy and efficiency.

Transformer-Based Models: Recent developments in deep learning have demonstrated remarkable performance in natural language processing tasks. Examples of these models are BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pretrained Transformer). These models' capacity to comprehend and produce text makes them potentially useful for phishing detection. In order to detect phishing efforts even more precisely, future study should investigate how transformer-based models can be modified to examine the textual content of URLs, emails, and web pages.

Hybrid Models: Creating hybrid models that incorporate the best features of several DL architectures is another exciting avenue to pursue. For instance, a hybrid model might include an LSTM to record temporal dependencies and a CNN to extract features from a URL's structure. These models have the potential to enhance performance by capitalizing on the complimentary advantages of disparate systems.

Attention Mechanisms: Attention mechanisms, which allow models to focus on the most relevant parts of the input data, have been shown to enhance the performance of DL models in various tasks. Incorporating attention mechanisms into phishing detection models could help them better identify key features that distinguish phishing URLs from legitimate ones, further improving detection accuracy.

2. Alternative Feature Engineering Techniques:

Feature engineering plays a critical role in the performance of phishing detection models. While this research introduced several novel features that improved model accuracy, there are still many unexplored avenues for feature engineering that could yield further improvements.

Contextual Features: Future research could investigate the use of contextual features that consider the broader environment in which a URL or email is encountered. For example, features that capture the typical behavior of a user or the reputation of a domain could provide additional context that helps distinguish phishing from legitimate activity.

Graph-Based Features: Graph-based features, which model the relationships between entities such as URLs, IP addresses, and domains, could offer new insights into phishing detection. Graph neural networks (GNNs) could be used to analyze these relationships, identifying patterns that are indicative of phishing campaigns.

Behavioral Features: Behavioral features, which capture how users interact with URLs and web pages, could also be valuable in phishing detection. For example, the time spent on a page or the sequence of clicks could provide clues about whether a user is being phished. Incorporating these features into phishing detection models could improve their ability to detect attacks that rely on social engineering.

3. Application of Transfer Learning:

Phishing detection could be greatly enhanced by transfer learning, which is modifying a model trained on one task to perform better on another related one. This method may be especially useful when there is a lack of tagged phishing data.

Domain Adaptation: One area of research could focus on domain adaptation, where a model trained on data from one domain (e.g., email phishing) is adapted to perform well on data from another domain (e.g., social media phishing). By leveraging knowledge from related domains, transfer learning could improve the robustness and generalizability of phishing detection models.

Cross-Lingual Phishing Detection: Phishing attacks are not limited to any single language or region. Transfer learning could be used to adapt phishing detection models to work across different languages and regions, making them more effective in global contexts. This could involve training models on multilingual datasets or using techniques like cross-lingual embeddings.

4. Reducing Computational Requirements of DL Models:

Although deep learning models have a high degree of accuracy, their computing demands may prevent them from being widely used. Subsequent studies might concentrate on creating methods to lessen these models' computational load without compromising their accuracy.

Model Compression: Techniques such as model pruning, quantization, and knowledge distillation can be used to reduce the size and complexity of DL models. These techniques remove redundant parameters or compress the model’s representation, resulting in faster inference times and lower resource consumption.

Efficient Architectures: The development of more efficient DL architectures, such as MobileNet or EfficientNet, which are designed to perform well on resource-constrained devices, could also be explored. These architectures could make it feasible to deploy DL-based phishing detection models on mobile devices, IoT devices, or edge computing platforms.

Distributed and Federated Learning: Distributed learning, where the computational workload is spread across multiple devices, and federated learning, where models are trained across decentralized devices while preserving data privacy, could also be explored. These approaches could make it possible to deploy DL models in environments with limited computational resources.

5. Addressing Evolving Phishing Techniques:

Phishing strategies are always changing because attackers are always coming up with new ways to get around security measures. To remain abreast of these changing risks, constant research and adaptation are needed.

Adversarial Learning: Adversarial learning, where models are trained using data that mimics the tactics used by attackers to evade detection, could be a valuable approach. By incorporating adversarial examples into the training process, phishing detection models can be made more resilient to evasion techniques.

Continuous Model Updating: To keep pace with evolving phishing tactics, phishing detection models need to be updated regularly. Research could explore the development of systems that can continuously learn from new data and adapt to emerging threats. This could involve the use of online learning or active learning techniques.

Integration with Threat Intelligence: Integrating phishing detection models with real-time threat intelligence feeds could help them stay updated with the latest phishing trends. By incorporating insights from threat intelligence, models could be better equipped to detect new types of phishing attacks as they emerge.

6. Expanding the Scope of Phishing Detection:

Finally, future research could explore expanding the scope of phishing detection beyond traditional email and web-based phishing.

Phishing on Social Media: With the growing popularity of social media platforms, phishing attacks have also targeted them. Since the content and interactions on social media platforms differ greatly from those on traditional web-based phishing, research might concentrate on creating models specifically made to identify phishing attempts on these platforms.

Phishing on mobile and IoT platforms: Phishing attacks are becoming more common as mobile devices and the Internet of Things (IoT) become more widely used. Subsequent investigations may delve into the creation of phishing detection models customized to the distinct attributes of mobile and Internet of things settings.

Voice and Video Phishing: Emerging technologies such as voice assistants and video conferencing tools are also potential targets for phishing attacks. Developing models capable of detecting phishing attempts in voice and video communications could be an important area of future research

# References

1. Suleman and A. A. Awan, "A novel machine learning based detection of phishing websites," in 2019 IEEE 2nd International Conference on Advanced Robotics and Mechatronics (ICARM), Toyonaka, Japan, 2019, pp. 522-527.
2. D. Thakur, K. Thakur, and N. Thakur, "Detection of phishing websites using decision tree algorithm," in Advances in Cyber Security and Computer Science, S. Singh et al., Eds. Springer, 2023, pp. 145-157.
3. A. Marakhimov, J. Ji, and J. Lee, "Random forest approach for phishing detection," in Proc. 14th Int. Conf. Security and Privacy in Communication Networks (SecureComm 2022), Singapore, 2022, pp. 112-124.
4. N. S. Shekokar, P. Thakur, and S. Chavan, "Machine learning and its application in phishing URL detection," Int. J. Comput. Appl., vol. 115, no. 3, pp. 26-30, Apr. 2015.
5. L. Jiang, Y. Wang, and S. Wei, "Phishing website detection based on support vector machine algorithm," in Proc. 2nd Int. Conf. Artificial Intelligence and Computer Engineering (AICE 2021), Beijing, China, 2021, pp. 57-62.
6. K. K. R. Choo, C. N. Liu, and J. Li, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," IEEE Commun. Surv. Tutor., vol. 19, no. 2, pp. 1327-1353, 2nd Quarter 2017.
7. S. Gopal, "Phishing website detection by machine learning techniques," GitHub, 2021. [Online]. Available: https://github.com/. Accessed: Aug. 27, 2024.
8. A. S. Kini, A. N. G. Reddy, M. Kaur, S. Satheesh, T. Martinetz, and H. Alshazly, "Ensemble deep learning and internet of things-based automated COVID-19 diagnosis framework," Contrast Media & Molecular Imaging, vol. 2022, Art. no. 7377502, 10 pages, 2022. [Online]. Available: <https://doi.org/10.1155/2022/7377502>.
9. A. Sharan, "Term co-occurrence and context window based combined approach for query expansion with the semantic notion of terms," Int. J. Web Sci., vol. 3, no. 1, pp. 1-12, 2017.
10. S. Kumar and S. K. Pathak, "A comprehensive study of XSS attack and the digital forensic models to gather the evidence," ECS Trans., vol. 107, no. 1, pp. 1-15, 2022.
11. C. S. Yadav et al., "Multi-class pixel certainty active learning model for classification of land cover classes using hyperspectral imagery," Electronics, vol. 11, no. 17, Art. no. 2799, 2022. [Online]. Available: <https://doi.org/10.3390/electronics11172799>.
12. C. S. Yadav et al., "Malware analysis in IoT & Android systems with defensive mechanism," Electronics, vol. 11, no. 15, Art. no. 2354, 2022. [Online]. Available: <https://doi.org/10.3390/electronics11152354>.
13. A Goswami, D Sharma, H Mathuku, SMP Gangadharan, CS Yadav, “Change Detection in Remote Sensing Image Data Comparing
14. J. Singh, "An efficient deep neural network model for music classification," Int. J. Web Sci., vol. 3, no. 3, pp. 34-45, 2022.
15. V. K. Bohat, "Neural network model for recommending music based on music genres," in Proc. 10th IEEE Int. Conf. Computer Communication and Informatics (ICCCI 2021), Coimbatore, India, Jan. 2021, pp. 1-6.
16. J. Singh, "Learning-based driver drowsiness detection model," in Proc. 3rd IEEE Int. Conf. Intelligent Sustainable Systems (ICISS 2020), Palladam, India, Dec. 2020, pp. 1163-1166.
17. A. Sharan, "Rank fusion and semantic genetic notion based automatic query expansion model," Swarm Evol. Comput., vol. 38, pp. 90-101, 2018.
18. R. Singh, "Ranks aggregation and semantic genetic approach based hybrid model for query expansion," Int. J. Comput. Intell. Syst., vol. 10, pp. 34-55, 2017.
19. A. Sharan, "A new fuzzy logic based query expansion model for efficient information retrieval using relevance feedback approach," Neural Comput. Appl., vol. 28, pp. 1-15, 2017.
20. C.-T. Lin et al., "IoT-based wireless polysomnography intelligent system for sleep monitoring," IEEE Access, vol. 6, pp. 55302-55312, Oct. 2017.
21. M. Prasad, Y. Daraghmi, P. Tiwari, P. Yadav, and N. Bharill, "Fuzzy logic hybrid model with semantic filtering approach for pseudo relevance feedback-based query expansion," in 2017 IEEE Symp. Series Computational Intelligence (SSCI), Honolulu, HI, USA, 2017, pp. 1-8.
22. R. Kumar, "Lexical co-occurrence and contextual window-based approach with semantic similarity for query expansion," Int. J. Intell. Inf. Technol. vol. 13, no. 3, pp. 57-78, 2020.

**Appendix :**

Open files from Google Drive:

from google.colab import drive

drive.mount('/gdrive')

%cd /gdriveOpen files from Google Drive

Adding form fields:

# @title Example form fields

# @markdown Forms support many types of fields.

no\_type\_checking = '' # @param

string\_type = 'example' # @param {type: "string"}

slider\_value = 142 # @param {type: "slider", min: 100, max: 200}

number = 102 # @param {type: "number"}

date = '2010-11-05' # @param {type: "date"}

pick\_me = "monday" # @param ['monday', 'tuesday', 'wednesday', 'thursday']

select\_or\_input = "apples" # @param ["apples", "bananas", "oranges"] {allow-input: true}

# @markdown ---