A Hybrid Deep Learning-Based Framework for Phishing Website Detection

by Yash V. Nadkarni

Honours Dissertation

Submitted in partial fulfilment for the degree of
BSc (Hons) Computer Science (Cyber Security)

Supervised by Dr. MD. Azher Uddin

Heriot-Watt University
School of Mathematical and Computer Sciences

18th November 2025

Authorship Declaration

Course code and name	F20PA - Research Methods and Requirements Engineering
Type of assessment	Individual
Coursework title	A Hybrid Deep Learning-Based Framework for Phishing Website Detection
Student name	Yash V. Nadkarni
Student ID	H00410472

By signing this form:

- I declare that the work I have submitted for individual assessment or the work I have contributed to a group assessment is entirely my own. I have not taken the ideas, writings or inventions of another person and used them as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements sections.

- I confirm that I have read, understood and followed the University's Regulations on academic misconduct as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism.

Student signature: Yash V. Nadkarni

Date: 18th November 2025

Your work will not be marked if a signed copy of this form is not included with your submission.

Acknowledgements

I would first like to give my deepest thanks to my supervisor, Dr. MD. Azher Uddin for his constant support and guidance throughout the development of this project, without which this would not have been possible.

I would like to extend my thanks to my close friends and family for providing me with their love and support through the hardest of times ensuring that I give this project my best.

Lastly, I would like to give my thanks to the rest of the Heriot-Watt University professors and faculty for providing me with the skills and foundation to pursue this project.

Abstract

The rampant growth of online-based services has led to an escalated dependence on the internet for daily communication, finance and more. This dependence has significantly increased the prevalence of phishing attacks that exploit human trust and technical vulnerabilities to steal sensitive personal information. Modern phishing techniques have become more sophisticated and complex, resulting in traditional detection systems struggling to adapt to this more dynamic and adaptive evolution, often failing to generalize past known patterns. Existing studies do not take into account and incorporate sequential temporal dependencies within their phishing detection systems. To address this prevailing limitation, a novel hybrid deep learning framework proposed by this dissertation aims to enhance the accuracy and efficiency of detecting phishing websites by employing a 1-Dimensional Convolutional Neural Network (1D-CNN), integrated with a Bidirectional Gated Recurrent Unit (BiGRU). Together, the framework is able to obtain higher-level patterns from URLs, along with any bidirectional temporal dependencies between them, thus allowing the model to capture complex patterns and correlations from the URLs. Utilizing two state-of-the-art datasets ensures the framework is trained on a diverse collection of URLs to improve consistency. Through rigorous testing and evaluations, the proposed hybrid framework seeks to contribute to the development of more adaptive and durable phishing detection systems for the protection of user integrity in this every evolving cyberspace.

Keywords: Phishing Detection, Phishing Websites, URLs, Hybrid Framework, Deep Learning, 1-Dimensional Convolutional Neural Network (1D-CNN), Bidirectional Gated Recurrent Unit (BiGRU), Framework Evaluation.

Table of Contents

1 Introduction

In the modern digital age, day-to-day life has become completely intertwined with the internet, which now serves as a global hub for all things pertaining to communication, finance, information and much more [3]. Most individuals and organizations now rely heavily on the internet for their daily social or productive activities such as social media, banking, education, entertainment, healthcare, etc. However, this evolution in technology has also seen an evolution in cyber threats from malicious actors who seek to exploit online vulnerabilities to compromise privacy and integrity. They utilize several techniques of cybercrime to carry this out, thus disrupting the daily flow of life online [24].

1.1 Background and Motivation

Phishing is a common term used in the digital age that is loosely derived from the process of 'luring' in the fishing activity. [11] describes it as the practice of deceiving users by masquerading malicious sources as legitimate. These sources, through which such a scam is usually carried out include websites, emails, SMS, etc. The attacker masks these websites and emails to impersonate legitimate companies such as government portals, medical sites, banks and finance websites and more. The main purpose of this cyberattack is to extract personal sensitive information from victims like passwords which are then used by the attacker for their own benefit [24]. In certain cases, these phishing attacks deceive victims into unknowingly installing malware and viruses onto their computers making them vulnerable to other forms of cyber-attacks.

Over the years, different forms of phishing attacks have been developed by hackers [3]. The common denominator in these attacks is the luring of victims to access a phishing site. Spoofing Emails are malicious emails sent by attackers that are written similarly to their legitimate counterparts with forged sender's address. Spear-Phishing is a form of Spoofing that targets specific individuals and organizations. Whaling is its subset that involves high ranking targets for obtaining classified information. SMiShing is an attack that uses SMS mobile services instead of emails while Vishing sees the attacker, pretending to be a high-ranking individual, scams the victim through a phone call.

According to [24], the main aspect of developing a phishing website, email, etc. is to make it resemble the official counterpart as close as possible using official logos and layouts so that the victim does not question its authenticity. [35] shows that there is one surefire way to tell whether a website is legitimate or not, and that is by its URL. A Uniform Resource Locator (URL) is the unique defining address of a specific resource available on the internet. Any malicious website seeking to impersonate its legitimate counterpart will have an irregular URL that will give away its legitimacy when examined by experienced internet users. An analysis shows that these URLs would include unnecessary special characters (e.g., @, /, -), or misspelled domain names (e.g., gooogle, fac3bo0k).

Unfortunately, due to the fast-moving lives of common folk, most people tend to not notice the irregular URLs. They're more focused on completing their everyday tasks, which makes them extremely vulnerable to phishing attacks. [25] showcases that during the Covid-19 pandemic, there was a huge surge in phishing attacks with its frequency increasing by 4 times during this period. [32] tells us that these phishing attacks used covid related domains to lure people by allegedly providing methods to stop the outbreak. According to the Anti-Phishing Working Group (APWG), their Phishing Activity Trends Report for 2025 [13] reveals that over 2 million phishing attacks have occurred in the first half of 2025 with the financial sector being frequently targeted.

As seen in Section 2, the integration of temporal dependencies modeling combined with extraction of higher-level feature patterns in phishing detection remains relatively unexplored. Despite CNN's ability to capture structural patterns of URLs, it is limited in learning temporal features. The proposed 1D-CNN-BiGRU hybrid framework addresses this limitation by being capable of capturing forward and backward temporal dependencies of the lexical features, thus allowing for detecting deeper relations and patterns of the URL's structure.

1.2 Aim and Objectives

Aim: To design, implement and evaluate a hybrid deep learning framework for the purpose of detecting phishing websites and assessing whether the hybrid model outperforms traditional ML based systems in terms of accuracy on standard phishing datasets.

Objectives:

(1) Acquire and preprocess publicly available phishing datasets.
(2) Develop a framework for a hybrid deep learning model.
(3) Utilize a 1-Dimensional Convolutional Neural Network (1D-CNN) for extracting lexical features and patterns.
(4) Utilize a Bidirectional Gated Recurrent Unit (BiGRU) to extract the temporal features and sequential dependencies.
(5) Train the hybrid framework on the preprocessed datasets.
(6) Evaluate the performance of the framework by means of accuracy, precision, recall and f1-score.
(7) Compare the evaluated performance of proposed framework against existing ML-based phishing detection models.

1.3 Organisation

The rest of this dissertation shall follow an organized structure. Section 2 presents the Background Research, where we shall be examining previous studies and research conducted. Section 3 shall cover the detailed description of the proposed hybrid framework with Section 4 covering the necessary requirements analysis for developing this framework. Section 5 will talk about the evaluation of the framework: the datasets on which it will be tested and the metrics on which it will be evaluated. The overall conclusions of these sections shall be discussed in Section 6. Lastly, the back matter shall contain the important Appendices covering the Gantt Chart and Risk Analysis as part of project management, as well as the Professional, Legal, Ethical and Social (PLES) considerations along with the list of references used.

2 Background Research

While increasing user's knowledge on phishing attacks is imperative, it is evident that it is not enough. The development of automated software capable of detecting and warning users of potential phishing attacks has seen significant growth overtime. This section shall cover the solutions proposed by previous studies conducted for detecting phishing sites. They are categorized based on the methodologies used.

2.1 List-Based Systems

Being the earliest form of anti-phishing protection, the core concept of list-based systems is the classification and storage of websites into white and black lists based on their legitimacy. Comparatively, these are simpler due to their fast lookups and require less system costs.

2.1.1 Whitelists. A whitelist is a list or database containing only those websites that are confirmed to be legitimate. When the user enters a site, the system cross-references the site's URL with the whitelist to see if the website is safe. If a website is not found, it is flagged as skeptical.

Earlier works include a system created by [9] that stores legitimate websites based off their IP addresses. Though the system provides a warning for unknown sites, it fails to classify new legitimate websites, showing the need to be constantly updated.

A popular system developed by [15] in 2016 that achieved an accuracy rate of 86.07%, includes the auto-updating of the whitelist on the client side to factor in newer legitimate websites. Their methodology included matching the IP address and extracting features from the website's source code.

2.1.2 Blacklists. Opposite a whitelist, a blacklist contains only the websites that are known to be malicious. Similar to how whitelists behave, a blacklist-based system cross-references the stored websites' features (e.g., URL, IP address) with those of the visited website. If a match is found, website access to the user is denied.

A zero-day phishing attack is a form of zero-day exploit [14] that involves a new phishing attack from a malicious website that has recently been created and is not yet known by blacklist systems. The period from the website's conception to its addition to a blacklist is called the Window of Vulnerability [10]. This type of attack is always resistant to blacklists no matter how frequently they are updated, thus highlighting the need for more adaptive systems. PhishTank [22] and Google Safe Browsing [8] are some of the most popular phishing black list services available.

2.2 Heuristics-Based Systems

The insufficiency of list-based systems made it evident that more dynamic solutions were needed. Heuristics-based systems [10], aim to extract a website's properties to determine its legitimacy. This can range from lexical URL attributes (e.g., @, /, -) to the HTML source code of the webpage.

[34] developed CANTINA which analyzed the webpage's text to extract keywords and terms. Using TF-IDF, the keywords were searched up in search engines for assessment.

If the website appeared in the results, it was deemed legitimate, else malicious. Despite exhibiting a high true positive rate, it suffers from being restricted to the English language due to limitations of TF-IDF. CANTINA+, a more enhanced system developed by [33] employs 8 more features based off visual similarities, HTML DOM and more. PhishNet from [23] which outperformed Google Safe Browsing [8], also uses heuristics through URL analysis to predict newer phishing sites.

Heuristics based systems laid the foundation for the transition towards Machine Learning based approaches that emphasize more on feature-based detections.

2.3 Visual Similarity-Based Systems

Another common approach to detect phishing websites is to analyze their screen shots to form predictions. These systems extract features from these webpages that alter the visual layouts and appearances. The system proposed by [16] extracts the position of elements, color schemes, CSS and HTML DOM trees, pixel features, etc. while another [31] employs a dual scale framework that analyzes global (layout) and local similarity (logos).

This methodology addresses zero-day attacks; however, it requires significant computational power for the processing of images. Furthermore, aesthetically different layouts due to bad CSS codes can result in a large percentage of false negatives [10].

2.4 Machine Learning-Based Systems

To further combat the evolving techniques used by phishing attacks, researchers have turned to ML based techniques to develop adaptive detection systems driven by data.

While heuristics systems rely on manually defined algorithms such as TF-IDF in the case of CANTINA [34], ML systems identify patterns and relations from data to form distinctions. This shift towards ML based systems marked a massive advancement that gave rise to superior predictive systems that are able to handle zero-day attacks.

ML models need appropriate datasets for proper training. As described by [32], feature selection a.k.a feature engineering is the method of selecting specific types of data that will aid the model in identifying patterns. For example, the number of '/' symbols in a URL is a feature. It is crucial to select only the right set of features for training the model, as using pointless or noisy features will result in higher computational costs and time which could harm the model's accuracy.

As the sites are classified as legitimate or phishing/malicious, these ML models are considered a binary classification problem. Over time, several different ML models have been developed with each one having different performance rates when it comes to the detection of phishing websites.

In 2013, one of the earliest studies involving ML based techniques for phishing detection saw [17] evaluating several different algorithms to test out their effectiveness. The dataset consisted of 37,000 phishing & legitimate URLs obtained from public whitelists and blacklists such as PhishTank [22]. After performing lexical analysis on, they then used these lexical features to train 4 different ML based models: Naïve Bayes, J48 Decision Tree, k-NN and SVM. Evaluation saw J48 Decision Tree outperforming the rest with a 93% accuracy.

In 2018, [29] performed a similar study where they evaluated the performance of Decision Trees, RF and GBM using various features extracted from URLs from PhishTank [22]. A technique called Principal Component Analysis (PCA) was applied to combine features to create new features. Upon evaluating their results, they found that the RF model achieved the highest accuracy rate of 98.40% on the PCA applied dataset.

A study from [19] proposed a multilayer stacked ensemble model which combines the predictive strengths of several ensemble ML algorithms to detect phishing sites. Ensemble & deep learning models such as MLP, kNN, RF, Logistic Regression and XGB were employed across 3 layers, with the outputs of one layer serving as inputs to the next, as seen in Fig.1. After evaluation, the model achieved a high accuracy (97.76%, 98.90%, 96.79%) across 3 different datasets. However, the employment of multiple ML algorithms simultaneously across several layers, results in incredibly high computational costs. Later studies indicate that the utilization of a single algorithm with proper feature selection is sufficient in achieving high accuracy.

The system proposed by [6] highlights the superiority of RF with an accuracy of 97.3% on a more popular UCI dataset [21], while another from [4] using a semantic features URL dataset [27] saw RF outperforming 15 other ML algorithms to achieve a 99.06% accuracy.

An experimental study by [1] examined the impact of dataset balancing, hyperparameter tuning and feature selection on the performance of phishing detection. Utilizing the UCI [21] and Mendeley 2018 [27] datasets, each model was trained and tested twice, once with and without optimization. For the UCI, fine-tuning improved accuracy, while it remained the same for Mendeley 2018. From this, it can be inferred that proper optimization can maintain or even improve accuracy while considerably reducing computation cost.

A study from [2] proposes an optimized stacking ensemble model for phishing detection using the UCI [21] Mendeley 2018 [27] and Mendeley 2020 [30] datasets. At first, RF & LightGBM had achieved the best accuracies, however, after applying Genetic Algorithm (GA)-based ensemble classifiers as seen in Fig.2, there was a shift in performance rankings. GA-based Gradient Boosting achieved the highest accuracy on UCI (97.13%), while GA-based XGBoost did the same for Mendeley 2018 and 2020 (98.57% & 97.35%). These results emphasize that despite RF performing strongly, feature selection and algorithm tuning can yield superior results.

One study [25] saw the proposal of a Rough Set Theory-Based Hybrid Feature Selection (RSTHFS) method to reduce the feature dependency in ML models. By combining Rough Set Theory (RST), Correlation-Based Feature Selection (CFS) and Cumulative Distribution Function (CDF), a feature cut-off is identified which reduced the feature count by 69.11% on 3 datasets [21], [30] and [27]. Evaluating the performance of the ML models, it was evident that RSTHFS was able to achieve similar accuracy while reducing runtime by 30%, highlighting the value of effective feature selection.

The diverse and more sophisticated nature of URLs from phishing attacks over the last couple years have compelled researchers to turn to Deep Learning (DL)-based systems. In contrast to ML models, DL models are able to extract more complex patterns from data that would otherwise require extensive human defined features. This is especially powerful in the defense against zero-day phishing attacks.

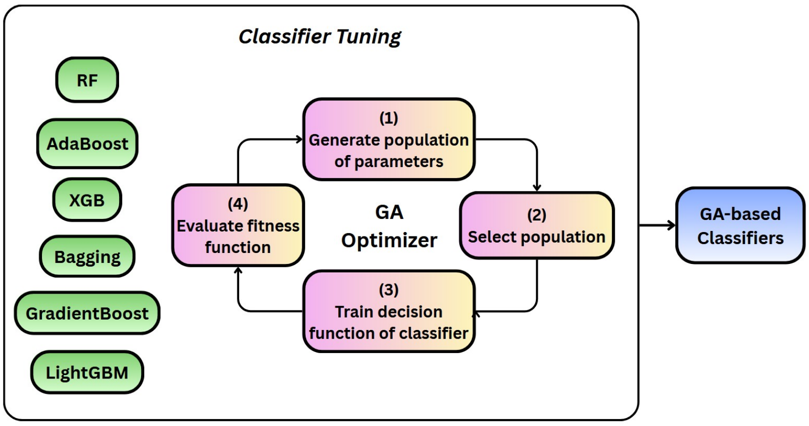

Fig. 1. Application of GA to ML classifiers by [2]

The study from [5] evaluated and compared traditional ML models against a DL model called Convolutional Neural Network (CNN) that extracts spatial patterns. The CNN trained on 20 epochs utilizing 3 dense layers with ReLU and sigmoid activation functions. Evaluation on 2 datasets [21], [30], saw it outperforming ML models with an accuracy of 99%.

Another study conducted by [7] analyzed and compared the performance of 3 DL models: CNN, LSTM and a LSTM-CNN hybrid system on the ISCX-URL2016 [12] dataset with CNN outperforming both LSTM and LSTM-CNN hybrid models by a difference of 3%, obtaining an accuracy of 99.2%. One study from [28] found that employing a Gated Recurrent Unit (GRU) to a Recurrent Neural Network (RNN) increased the accuracy from 74% to 99.18%, highlighting GRU's effectiveness.

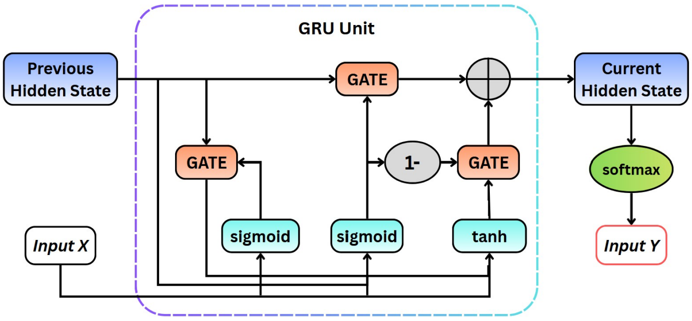

Fig. 2. Illustration of the architecture of a GRU Unit used by [28]

2.5 Critical Analysis of Related Work

Research on phishing detection has produced several methodologies, yet each exhibits its own limitation. List-based systems [9] demonstrate simple implementations but require constant updating making them ineffective against zero-day attacks. This highlights the need for adaptive methods over purely reactive systems. Systems analyzing visual features

[16] resolve this, but require complete webpage rendering, resulting in high computational costs in addition to a high rate of false negatives from bad designs [10]. Another attempt comes from heuristics systems [34], [33] and [23] using predefined algorithms and fixed heuristics. However, these suffer from hand-crafted logic that attackers can easily bypass by altering URL keywords and structure, making them lack adaptability to evolving strategies.

Adaptability through learning solves this limitation. Machine learning-based systems demonstrate a significant improvement over heuristics by learning statistical patterns from data instead of relying on predefined rules with ensemble methods such as Random Forest [29] [6] [4] consistently outperforming all other classifiers. However, their dependence on extensive manual feature engineering requires that features be repeatedly redesigned as phishing tactics evolve. In addition, features that are effective on one dataset may fail on another, highlighting the need for an even more adaptive approach that also performs consistently across different data. Complex ML systems [19] [2] show similar performances but incite high computational costs without resolving hand-crafted feature dependence. In that regard, deep learning systems prove to be more efficient than ML systems as they eliminate the need for manual feature engineering by learning representations and patterns directly from raw data. CNN models [5] [7] capture local spatial patterns while GRU models [28] extract temporal dependencies to improve accuracy. However, most deep learning models treat URL data as purely spatial or purely sequential, with each model struggling to capture information from the other type.

CNNs struggle to capture sequential relationships between paths, subdomains and queries while GRUs struggle to extract spatial cues such as suspicious characters. This highlights the lack of joint architecture combining both dependencies in phishing detection which this dissertation aims to address by proposing a hybrid framework utilizing a 1D-CNN and BiGRU offering improved adaptability with reduced reliance on manually-crafted features.

2.6 Comparison of Related Work

Study	Dataset	Model	Accuracy
[17]	PhishTank [22]	J48 Decision Tree	93%
[19]	UCI_2015 [21], Mendeley_2018 [27], Mendeley_2020 [30]	Multilayered Stacked Ensemble Model	98.90%
[4]	Mendeley_2018 [27]	Random Forest	99.06%
[2]	UCI_2015 [21], Mendeley_2018 [27], Mendeley_2020 [30]	GA-based XGBoost	98.57%
[7]	ISCX-URL2016 [12]	CNN	99.20%
[28]	Kaggle Dataset, PhishTank [22]	RNN-GRU	99.18%

Table 1. Comparison of accuracy from related studies.

3 Methodology

As illustrated in Fig. 4, the proposed phishing detection framework utilizes a hybrid approach employing a 1-Dimensional Convolutional Neural Network (1D-CNN) and a Bidirectional Gated Recurrent Unit (BiGRU) for classifying websites as phishing or legitimate. This 2-system approach is designed to utilize both the 1D-CNN's ability to capture local lexical patterns and the BiGRU's capability to model global sequential dependencies from the URL features, thus providing a novel, robust and effective phishing detection model.

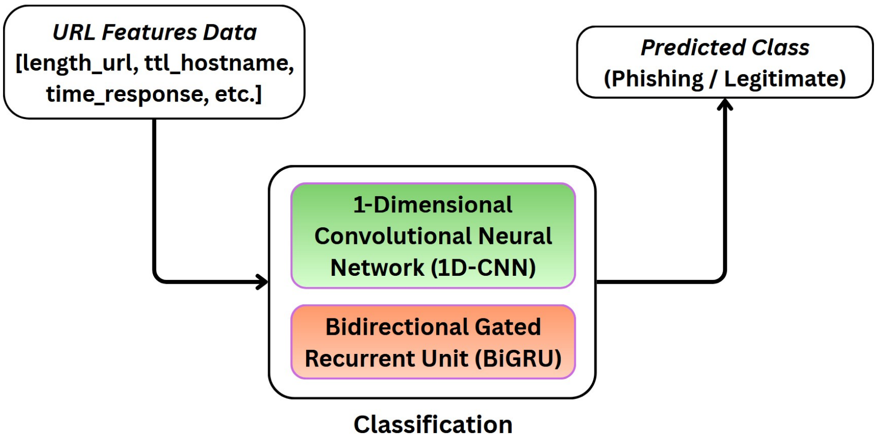
URL Features Data
[length_url, ttl_hostname, time_response, etc.]
Predicted Class
(Phishing / Legitimate)
1-Dimensional Convolutional Neural Network (1D-CNN)
Bidirectional Gated Recurrent Unit (BiGRU)
Classification

Fig. 3. Architecture of Proposed Framework

3.1 Convolutional Neural Network

The processing begins with the 1D-CNN, which functions to capture the lexical and structural patterns between the features of the URL. As stated in [20], CNNs make use of local connectivity, pooling and shared weights through multiple different layers to achieve this. In the proposed framework, several convolutional layers with varying filter sizes are employed to detect the local relations of features from one layer to the next. Each layer each is then followed by a non-linearity activation function and a max pooling layer to reduce dimensionality while still retaining important information.

Fig. 5 displays the visual representation of the architectural structure of a 1D-CNN employed by [18].

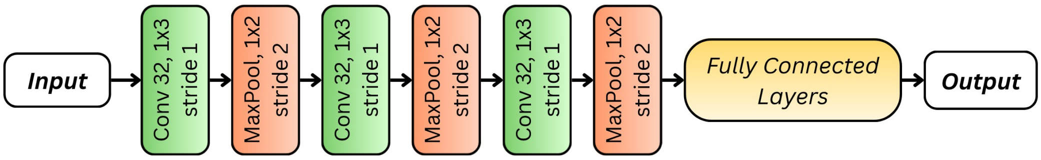

Fig. 4. Illustration of 1D-CNN model by [18]

3.2 Bidirectional Gated Recurrent Unit

The BiGRU layer extends the modelling capacity of the framework to further enhance performance. As per [18], a BiGRU model comprises of a forward directional and a backward directional GRU, which allows it to capture the contextual dependencies. [26] describes this as being able to capture elements from both the past and future in the data sequence. This bidirectional processing of the extracted feature maps from the 1D-CNN allows the proposed framework to ensure both the preceding and succeeding features in the URL influence the final outcome.

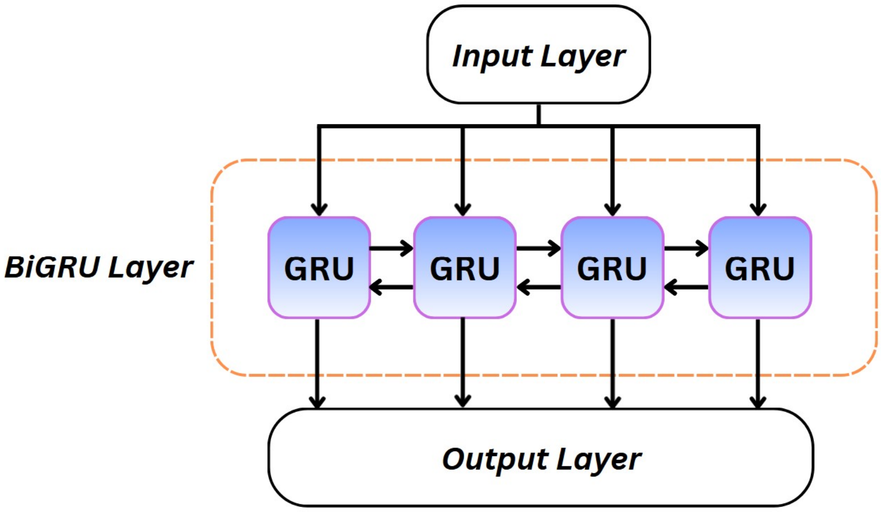

Fig. 5. One BiGRU Layer as depicted by [18]

The output feature vectors are then passed through the CNN's dense layers for obtaining the final classification. The employment of specific hyper parameters: sigmoid activation function in the output layer, adam optimizer and binary cross-entropy loss function ensures enhanced robustness and efficiency in obtaining a strong accuracy score with optimal performance.

4 Requirements Engineering

This section shall cover all the necessary requirements that will have to be undertaken in order to ensure the framework is successfully developed. The requirements shall be prioritized based on the MoSCoW method.

4.1 Functional Requirements

ID	Requirement	Priority
FR.1	Dataset Preprocessing	Must
FR.1.1	Datasets should be cleaned & normalized to remove redundancy and ensure better accuracy.	Must
FR.1.2	Datasets should be split into training and testing sets.	Must
FR.1.2.1	Utilize k-Fold Cross Validation for splitting the datasets.	Should
FR.1.3	Preprocessed datasets should be saved separately for easier access.	Should
FR.1.4	Utilizing RSTFHS to reduce the number of required features.	Should
FR.2	Implementation of Models	Must
FR.2.1	Training traditional ML models on the training set of our dataset.	Must
FR.2.2	Implementation and integration of 1D-CNN and BiGRU models, optimized with appropriate hyperparameters, to obtain desired hybrid framework.	Must
FR.2.3	Train the hybrid framework on the preprocessed train set.	Must
FR.2.4	Test the trained framework on the test set of the datasets.	Must
FR.3	Performance Evaluation	Must
FR.3.1	From the model's performance, generate the confusion matrix.	Must
FR.3.2	Calculate the empirical evaluation metrics using values derived from the confusion matrix.	Must
FR.3.3	Record the total training time for each model for analysis.	Should
FR.3.4	Identify which features influence the model's predictions the most.	Could
FR.4	Comparative Analysis	Must
FR.4.1	Compare & analyze the performance metrics of ML models against that of our proposed hybrid framework.	Must

FR.4.2	Compare the performance of models across the two datasets.	Must
FR.4.3	Compare the performance of models with and without feature selection & hyperparameter tuning.	Should
FR.5	Miscellaneous	Could
FR.5.1	Development of browser extension utilizing the hybrid framework.	Could

Table 2. Table of Functional Requirements.

4.2 Non-Functional Requirements

ID	**Requirement**	**Priority**
NFR.1	Hybrid framework must obtain a high accuracy >= 80%.	M
NFR.2	Access to models and datasets shall be restricted to the author and supervisor only.	S
NFR.3	Datasets must be stored and processed locally.	M
NFR.4	Throughput during testing phase should be tolerable.	S

Table 3. Table of Non-Functional Requirements.

4.3 Hardware & Software Requirements

ID	**Component**	**Requirement**
HR.1	Processor (CPU)	Intel i5 or AMD Ryzen 5
HR.2	Graphics Card (GPU)	Any NVIDIA GPU with 4+ GB VRAM
HR.3	Memory (RAM)	8 GB
HR.4	Storage	10 GB SSD for fast processing
SR.1	Software	• Python 3.9+ • TensorFlow / Keras 2.12+ • NumPy 1.25+ • Pandas 2.0+ • Scikit-learn 1.3+ • Matplotlib

Table 4. Table of Hardware & Software Requirements.

5 Evaluation

5.1 Dataset

For training and evaluating the proposed hybrid framework, two publicly available benchmark state-of-the-art datasets shall be employed, the UCI Phishing Websites Dataset [21] and the Mendeley 2020 Dataset [30]. Utilizing two datasets ensures that our framework performs consistently through evaluation across URL data of varying size and complexities. Both datasets comprise of previously extracted features from existing phishing and legitimate websites, thus eliminating the need for a feature extraction phase.

5.1.1 UCI 2015 Dataset. :

The University of California Irvine Machine Learning Repository contains the Phishing Websites dataset which is one of the most widely used datasets in the field of phishing detection [21]. The dataset comprises 11,055 websites (4898 phishing, 6157 legitimate), each one distinguished by 30 different handcrafted features describing the website's URL's structural and lexical attributes. The 31st feature labels the website as phishing or legitimate, thus providing a clear binary classification problem.

5.1.2 Mendeley 2020 Dataset. :

With contrast to the UCI dataset, the Mendeley 2020 Dataset [30] comprises of 111 features extracted from 88,647 websites (30,647 phishing, 58,000 legitimate). The features extracted are more comprehensive encompassing the URL's lexical properties (length, entropy), domain attributes (age, DNS) and network attributes (IP, ssl validity). The proposed hybrid framework based on deep learning models will surely benefit from the well-suited size and variety of this dataset.

5.2 Evaluation Metrics

From testing the hybrid framework, 4 metrics can be obtained directly which are then utilized to generate the confusion matrix.

- **True Positives (TP):** The number of phishing websites correctly classified by the framework.
- **True Negatives (TN):** The number of legitimate websites correctly classified by the framework.
- **False Positives (FP):** The number of legitimate websites incorrectly classified as phishing by the framework.
- **False Negatives (FN):** The number of phishing websites incorrectly classified as legitimate by the framework.

From the confusion matrix, we can derive several performance metrics that will precisely indicate how effective the framework is and where it stands. These metrics are based on the standard empirical classification metrics used to evaluate performance.

	Phishing	**Legitimate**
Phishing	TP	FP
Legitimate	FN	TN

Table 5. Confusion Matrix from testing the hybrid framework.

5.2.1 Accuracy. :

Proportion of correctly classified websites among all predictions. It represents the overall performance of the hybrid framework in phishing detection in balanced datasets. However, both datasets that the hybrid framework will utilize are imbalanced with legitimate websites dominating. Thus, additional metrics are needed.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

5.2.2 Precision. :

Proportion of correctly classified phishing websites among all websites classified as phishing by the framework. A high precision indicates that the model produced very few false positives, thus preventing the unnecessary blacklisting of legitimate websites.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

5.2.3 Recall (Sensitivity). :

Proportion of correctly classified phishing websites among all phishing websites. It can be seen as the accuracy of correctly predicting only phishing websites. A high recall indicates that the framework is able to detect and capture most phishing attacks, thus reducing the chances of letting real threats slip.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

5.2.4 F1-Score. :

Represents the harmonic mean of the precision and recall. It balances FP and FN and is especially useful for imbalanced datasets.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

6 Conclusions

In conclusion, this dissertation proposes a hybrid deep learning framework to address an existing research gap in phishing websites detection. The proposed hybrid framework employs and integrates a 1-Dimensional Convolutional Neural Network (1D-CNN) with a Bidirectional Gated Recurrent Unit (BiGRU).

The research began with an extensive literature review (refer to Section 2), examining existing detection methodologies and datasets and highlighting their efficiency in detecting phishing websites. Through this, the scarcity of models combining temporal dependencies with lexical features extraction was made apparent (research gap). As such, an architecture for a hybrid deep learning framework (refer to Section 3) was conceived to address this research gap. This architecture sees the employment of a 1D-CNN to extract the local lexical features from the URLs of phishing websites, while the employment of a BiGRU component captures the temporal dependencies of these lexical URL tokens for better understanding the sequential relations between them. Together, these models form the end-to-end hybrid framework capable of detecting and identifying phishing websites.

To certify that, following the implementation, the framework shall be trained and tested on popular publicly available phishing datasets to ensure consistency, robustness and scalability. Empirical classification metrics shall be employed to evaluate the framework's performance, to then be compared against existing ML models to identify its standing amongst them (refer to Section 5).

Looking forward, future work shall focus primarily on implementing the 1D-CNN and BiGRU models and integrating them to develop the proposed hybrid framework. This shall include training, hyperparameter tuning and feature extraction. As mentioned above, the framework will then be tested and compared to achieve our ultimate aim of developing a robust and adaptive phishing detection system for contributing to a more secure and safer cyberspace.

References

[1] S.R. Abdul Samad, S. Balasubaramanian, A.S. Al-Kaabi, B. Sharma, S. Chowdhury, A. Mehbodniya, J.L. Webber, and A. Bostani. 2023. Analysis of the performance impact of fine-tuned machine learning model for phishing URL detection. *Electronics* 12(7) (2023), p.1642.

[2] M. Al-Sarem, F. Saeed, Z.G. Al-Mekhlafi, B.A. Mohammed, T. Al-Hadhrami, M.T. Alshammari, A. Alreshidi, and T.S. Alshammari. 2021. An optimized stacking ensemble model for phishing websites detection. *Electronics* 10(11) (2021), p.1285.

[3] M.M. Ali and N.F. Mohd Zaharon. 2024. Phishing—A cyber fraud: The types, implications and governance. *International Journal of Educational Reform* 33(1) (2024), pp.101–121.

[4] A. Almomani, M. Alauthman, M.T. Shatnawi, M. Alweshah, A. Alrosan, W. Alomoush, and B.B. Gupta. 2022. Phishing website detection with semantic features based on machine learning classifiers: a comparative study. *International Journal on Semantic Web and Information Systems (IJSWIS)* 18(1) (2022), pp.1–24.

[5] N.F. Almujahid, M.A. Haq, and M. Alshehri. 2024. Comparative evaluation of machine learning algorithms for phishing site detection. *PeerJ Computer Science* 10 (2024), p.2131.

[6] S. Alnemari and M. Alshammari. 2023. Detecting phishing domains using machine learning. *Applied Sciences* 13(8) (2023), p.4649.

[7] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q.E.U. Haq, K. Saleem, and M.H. Faheem. 2023. A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. *Electronics* 12(1) (2023), p.232.

[8] Google Safe Browsing. n.d. Google Safe Browsing: keeping over five billion devices safer [online]. Available at: https://safebrowsing.google.com/. (Accessed: 2025-10-17).

[9] Y. Cao, W. Han, and Y. Le. 2008. Anti-phishing based on automated individual white-list. In *Proceedings of the 4th ACM workshop on Digital identity management.* pp.51–60.

[10] C.M.R. da Silva, E.L. Feitosa, and V.C. Garcia. 2020. Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. *Computers Security* 88 (2020), p.101613.

[11] A. Fedele, M. Tonin, and M. Valerio. 2024. Phishing attacks: An analysis of the victims' characteristics based on administrative data. *Economics Letters* 237 (2024), p.111663.

[12] Canadian Institute for Cybersecurity. 2016. URL dataset (ISCX-URL2016) [dataset]. Available at: https://www.unb.ca/cic/datasets/url-2016.html. (Accessed: 2025-10-18).

[13] Anti-Phishing Working Group. 2025. Phishing Activity Trends Report – 2nd Quarter 2025. Available at: https://docs.apwg.org/reports/apwg_trends_report_q2_2025.pdf. (Accessed: 2025-10-16).

[14] M. Guo, G. Wang, H. Hata, and M.A. Babar. 2021. Revenue maximizing markets for zero-day exploits. *Autonomous Agents and Multi-Agent Systems* 35(2) (2021), p.36.

[15] A.K. Jain and B.B. Gupta. 2016. A novel approach to protect against phishing attacks at client side using auto-updated white-list. *EURASIP Journal on Information Security* 2016(1) (2016), p.9.

[16] A.K. Jain and B.B. Gupta. 2017. Phishing detection: analysis of visual similarity based approaches. *Security and Communication Networks* 2017(1) (2017), p.5421046.

[17] J. James, L. Sandhya, and C. Thomas. 2013. Detection of phishing URLs using machine learning techniques. In *2013 international conference on control communication and computing (ICCC).* pp.304–309.

[18] J.B. Joolee, M.A. Uddin, and S. Jeon. 2022. Deep multi-model fusion network based real object tactile understanding from haptic data. *Applied Intelligence* 52(14) (2022), pp.16605–16620.

[19] L.R. Kalabarige, R.S. Rao, A. Abraham, and L.A. Gabralla. 2022. Multilayer stacked ensemble learning model to detect phishing websites. *IEEE Access* 10 (2022), pp.79543–79552.

[20] Y. LeCun, Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521(7553) (2015), pp.436–444.

[21] R. Mohammad and L. McCluskey. 2015. Phishing Websites. UCI Machine Learning Repository. Available at: https://doi.org/10.24432/C51W2X. (Accessed: 2025-10-19).

[22] PhishTank. n.d. PhishTank: a free community site for submitting and verifying suspected phishing websites and emails [online]. Available at: https://www.phishtank.com/. (Accessed: 2025-10-17).

[23] Kumar M. Kompella R.R. Prakash, P. and M. Gupta. 2010. Phishnet: predictive blacklisting to detect phishing attacks. In *2010 Proceedings IEEE INFOCOM*. pp.1–5.

[24] O.K. Sahingoz, E. Buber, O. Demir, and B. Diri. 2019. Machine learning based phishing detection from URLs. *Expert Systems with Applications* 117 (2019), pp.345–357.

[25] J.H. Setu, N. Halder, A. Islam, and M.A. Amin. 2025. RSTHFS: A Rough Set Theory-Based Hybrid Feature Selection Method for Phishing Website Classification. *IEEE Access* 13 (2025), pp.68820–68830.

[26] D. She and M. Jia. 2021. A BiGRU method for remaining useful life prediction of machinery. *Measurement* 167 (2021), p.108277.

[27] C.L. Tan. 2018. Phishing dataset for machine learning: Feature evaluation. Mendeley Data, 1(8)., V1. Available at: https://doi.org/10.17632/h3cgnj8hft.1. (Accessed: 2025-10-19).

[28] L. Tang and Q.H. Mahmoud. 2021. A deep learning-based framework for phishing website detection. *IEEE Access* 10 (2021), pp.1509–1521.

[29] I. Tyagi, J. Shad, S. Sharma, S. Gaur, and G. Kaur. 2018. A novel machine learning approach to detect phishing websites. In *2018 5th International conference on signal processing and integrated networks (SPIN)*. pp.425–430.

[30] G. Vrbančič. 2020. Phishing Websites Dataset. Mendeley Data, V1. Available at: https://doi.org/10.17632/72ptz43s9v.1. (Accessed: 2025-10-19).

[31] M. Wang, L. Song, L. Li, Y. Zhu, and J. Li. 2024. Phishing webpage detection based on global and local visual similarity. *Expert Systems with Applications* 252 (2024), p.124120.

[32] Y. Wei and Y. Sekiya. 2022. Sufficiency of ensemble machine learning methods for phishing websites detection. *IEEE Access* 10 (2022), pp.124103–124113.

[33] G. Xiang, J. Hong, C.P. Rose, and L. Cranor. 2011. Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)* 14(2) (2011), pp.1–28.

[34] Y. Zhang, J.I. Hong, and L.F. Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web*. pp.639–648.

[35] R. Zieni, L. Massari, and M.C. Calzarossa. 2023. Phishing or not phishing? A survey on the detection of phishing websites. *IEEE Access* 11 (2023), pp.18499–18519.

A Project Management

In order to ensure the research is consistent with the objectives and is completed on time without being rushed, proper planning and organization is required from the start until the end. This section outlines the scope and planning of the projects along with the placement of milestones to ensure progress is executed smoothly. Moreover, the section shall also contain an analysis of possible risks that can be encountered during the project, and ways to mitigate them.

A.1 Project Scope & Deliverables

This research aims to address the current limitation of phishing websites detection systems, specifically the lack of existing systems that combine lexical feature extraction from phishing URLs with temporal sequential dependencies to understand patterns and relations. To overcome this gap, this project proposes an end-to-end hybrid deep learning framework, integrating a 1-Dimensional Convolutional Neural Network (1D-CNN) for extracting lexical features, with a Bidirectional Gated Recurrent Unit (BiGRU) for capturing sequential dependencies.

The project scope can be divided into the following key tasks for its development:

(1) Preprocessing publicly available phishing datasets.
(2) Normalizing data for better training compatibility.
(3) Implementing the 1D-CNN to extract lexical features.
(4) Integrating a BiGRU model to extract temporal features.
(5) Combining both models to form the hybrid framework.
(6) Evaluating the performance of the framework using empirical metrics.
(7) Comparing the performance of the framework against existing ML based systems.
(8) Documenting possible risks that can be encountered and providing mitigation strategies.

Project Deliverables:

The project is divided into 5 deliverables, each of which has to be submitted before their specific deadlines. It is imperative for all tasks pertaining to each deliverable to be completed in an organized and timely fashion for smooth delivery. The deliverables are described below.

A.1.1 Project Proposal and Research Document (Semester 1). :

The first deliverable is the research document establishing the foundation of the project by highlighting a brief overview of the study and the preparation done for it. It shall state the project's aim and objectives concisely along with the background behind the motivation of the project. A comprehensive literature review of existing studies and research in the field of phishing detection systems shall be conducted. The methodology behind the project's proposed framework shall be established, along with the datasets and empirical metrics on which it will be evaluated. This report will also present a work plan with milestones, possible risks associated with the project and provide strategies to mitigate them, as well as address any relevant professional, legal, ethical and social issues.

A.1.2 5-Minute Feasibility Video (Semester 1). :

The second deliverable is a short 5 minute video and presentation explaining the work done behind the overview and preparation for this project as well as an explanation of the feasibility of the proposed 1D-CNN-BiGRU hybrid framework's completion within planned timeframes.

A.1.3 Dissertation Report (Semester 2). :

The third deliverable is the final comprehensive dissertation, which documents the complete lifecycle of the project's implementation including detailed descriptions of the methodology, systems and algorithms. It shall also include the detailed evaluation process, along with an in-depth analysis of the performance evaluated against other models to establish its standing and contributions.

A.1.4 Project Management Repository (Semester 2). :

The fourth deliverable is a project management repository containing all project related materials including: relevant research papers, datasets, diagrams and figures, logbooks, documentations and source code files. The repository shall be maintained and updated throughout the development lifecycle in order to provide traceability as well as act as evidence of work.

A.1.5 Q&A Session (Semester 2). :

The final deliverable is a Q&A session between the author of this dissertation and the supervisor with an additional marker. It is here that the author's knowledge and capabilities on phishing detection systems with regards to the project's design and framework implementation will be tested.

A.2 Project Plan & Gantt Chart

The project plan serves as a structured timeline depicting the execution of every task from the beginning of the project's preliminary research, through to the end of every necessary deliverable. It highlights the important tasks, milestones and deadlines that need to be met for smooth completion of the project. The plan is divided into two semesters, each corresponding to the two main deliverables required. The Gantt charts shown in Figures 7-11 depict the planning and listing of tasks required for completion of each deliverable for each semester in time phased manner.

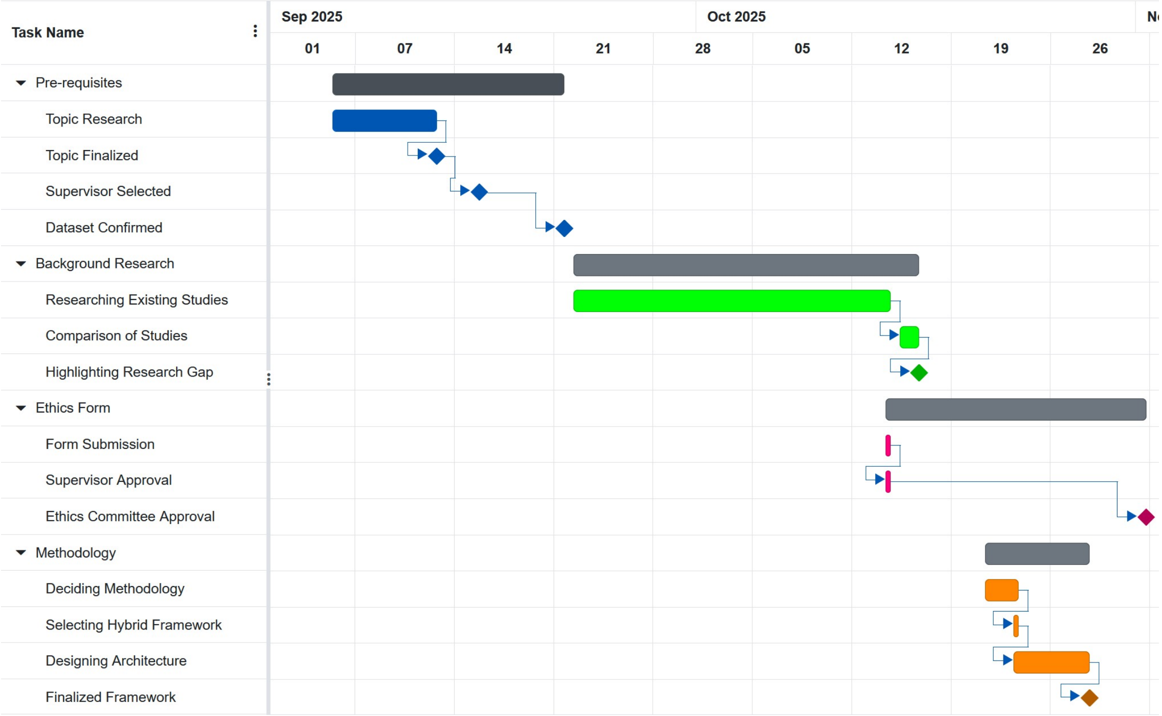

Fig. 6. Project Timeline for Semester 1 - Part 1

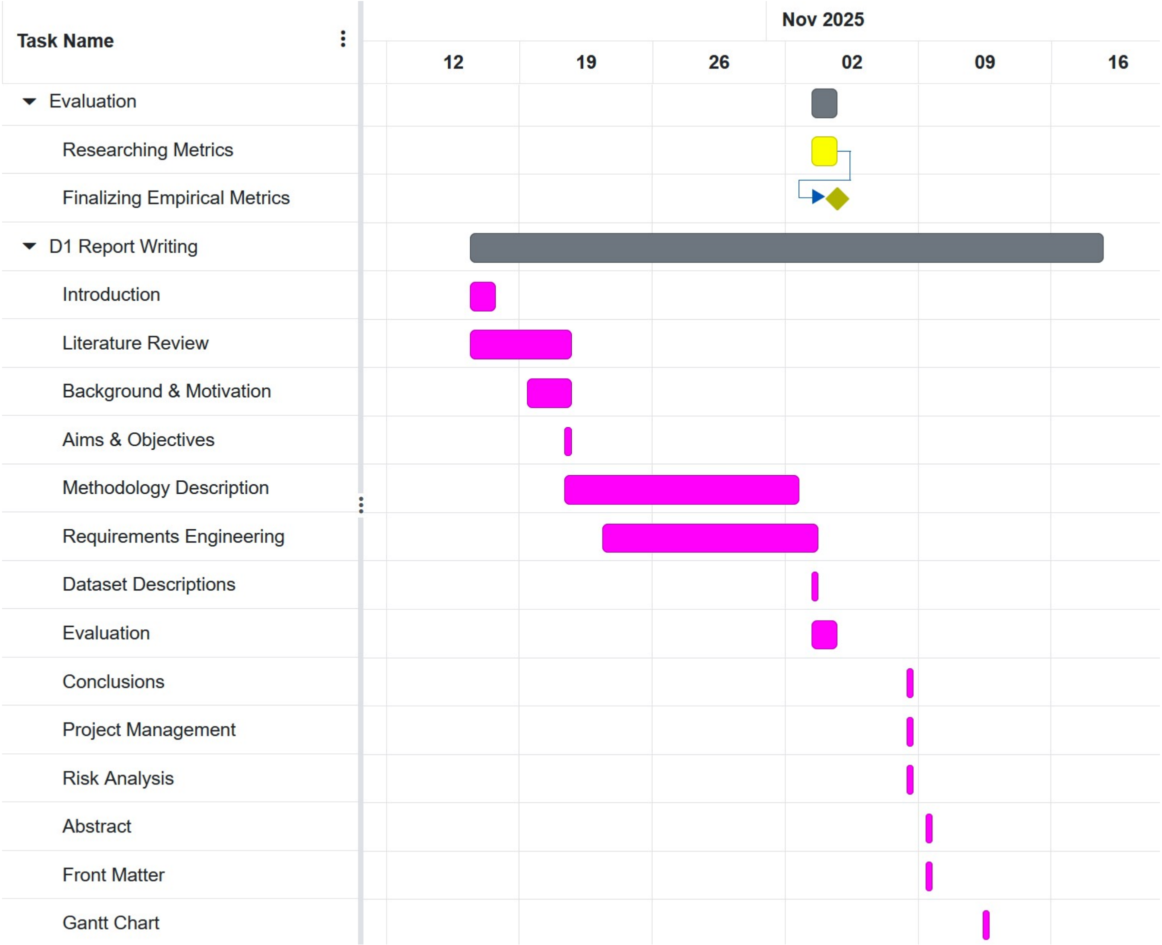

Fig. 7. Project Timeline for Semester 1 - Part 2

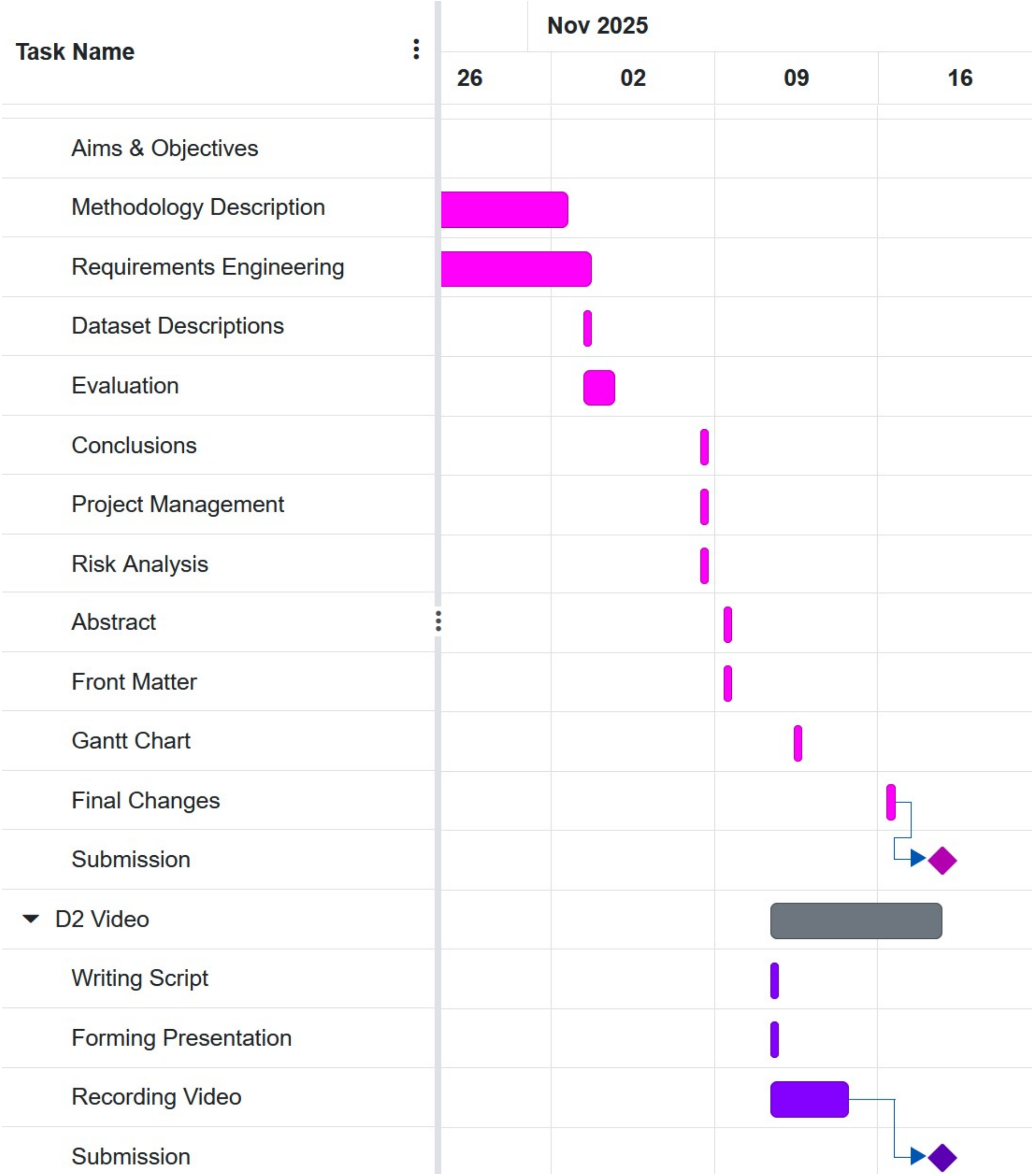

Fig. 8. Project Timeline for Semester 1 - Part 3

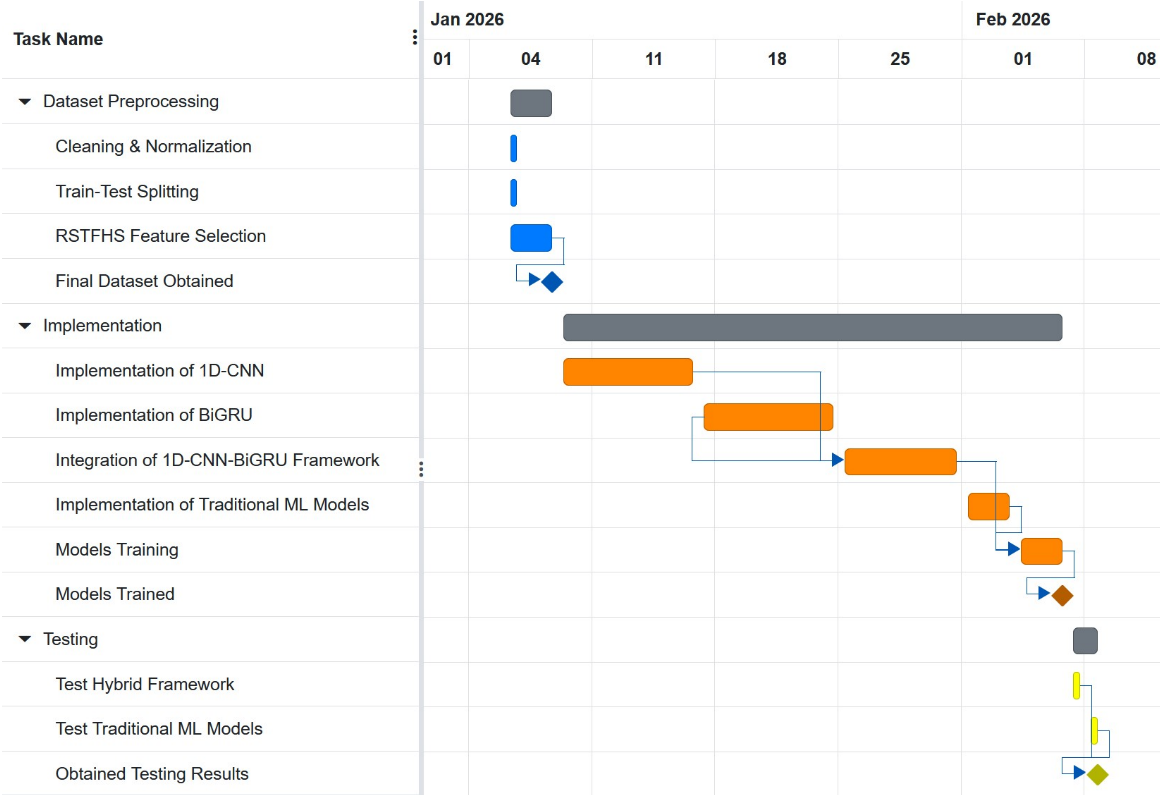

Fig. 9. Project Timeline for Semester 2 - Part 1

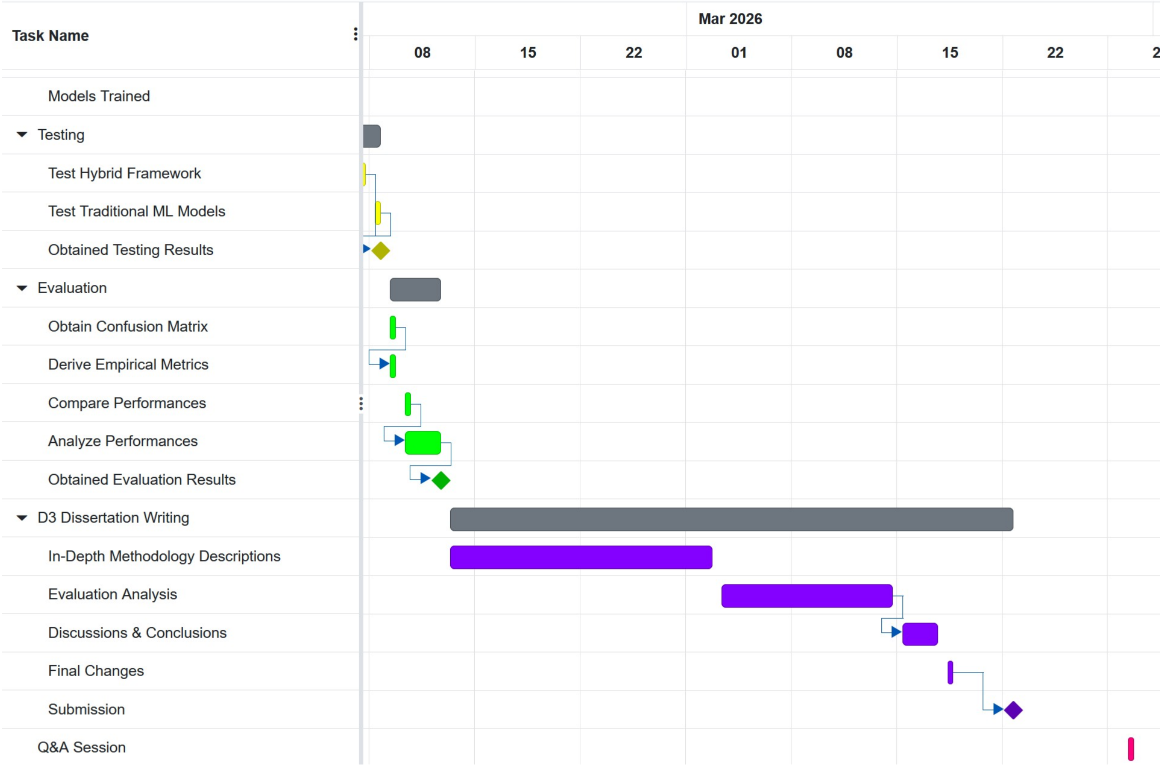

Fig. 10. Project Timeline for Semester 2 - Part 2

A.3 Risk Analysis

This section shall go over the possible risks that can be encountered during the development of the project. It is imperative that all possible risks are conceptualized as part of risk analysis so as to ensure that the project proceeds on schedule without any unforeseen complications. Table. 6 outlines the most probable risks, their likelihood and impact, as well as a possible strategy to mitigate or eliminate it. The likelihood can be classified as Very Likely, Likely, Possible and Unlikely. Impact can be classified as Very High, High, Medium and Low.

Risk	**Likelihood**	**Impact**	**Mitigation Strategy**
Low performance of hybrid framework.	Likely	High	Further hyper-parameter tuning, ensuring model is properly coded
Hybrid framework does not outperform other ML models.	Very Likely	High	Extensive hyper-parameter tuning.
Overfitting During Training	Likely	Medium	Implement cross validation, dropout or early stopping.
Inconsistent Evaluation Results.	Possible	High	Conduct multiple test runs and ensure consistent implementation of empirical metrics.
Data Corruption / Loss of Progress	Unlikely	Very High	Make regular copies and backups.
System Failure	Unlikely	Very High	Make backup saves to cloud storage.
Insufficient Hardware Capabilities & Resources	Unlikely	High	Utilize cloud-based environments such as Google Colab.
Time Constraints	Likely	Medium	Adhere to the planned timeline in the Gantt charts.
Bugs in developed code	Likely	Medium	Debugging and testing every feature in increments.

Table 6. Risk Analysis Table.

B Professional, Legal, Ethical and Social Issues

This project aligns with the standards and principles set by the institution and certified computing bodies to ensure responsible academic integrity and provide benefit to society.

B.1 Professional Issues

This project upholds strong standards in the procedures for collecting, preprocessing, evaluating and reporting on appropriate data. All systems, frameworks and models utilized follow current modern methodologies in machine learning and cybersecurity. This projects relies only on employing well established open source tools such as Python, Scikit-Learn, TensorFlow, Keras, Pandas, NumPy, Matplotlib and any other library that may be relevant to detecting phishing websites, which are referenced within their terms of use. Furthermore, the professional procedures for ensuring and maintaining integrity and competence though development and documentation are consistent with the standards set by the British Computing Society (BCS).

B.2 Legal Issues

This project complies with relevant data protection laws, including regulations set by the United Arab Emirates and General Data Protection Regulation (GDPR) for the handling of datasets. The phishing datasets employed in this project consist of publicly available technical URL features, none of which contain any user-specific personal information. These datasets are utilized solely for the purpose of academic research under institutional supervision. Any performance data along with its evaluation, is stored locally and will not be accessed by anyone other than the author and project supervisor.

B.3 Ethical Issues

This project does not require nor does it utilize any form of human participation for its purpose. As such, no sensitive personal information pertaining to specific individuals is stored or processed. The datasets employed contain only technical URL features of publicly available phishing and legitimate websites, with no information that can trace back to any individual. The evaluation of the proposed framework's performance shall be fairly conducted and validated using empirical metrics only. This project respectfully adheres to the data management requirements set by Heriot-Watt University's ethics committee.

B.4 Social Issues

The purpose of this project is to contribute positively to the societal efforts against phishing detection by proposing a hybrid deep learning framework, capable of detecting and classifying malicious phishing websites with an increased accuracy and precision. The state-of-the-art datasets are widely used and recognized, containing only technical URL features and no human data. The project and its evaluation are purely technical without introducing or pertaining to any other broader sensitive social issue. Any correlation found is purely coincidental and unintentional.