31482 Honours Project

Assessment Task 2: Research Report and Research Work

Name: Hugh Riddle – 14241323

Degree: Bachelor of Computing Science (Honours)

Research Topic: Fine-Tuned Lexical URL Phishing Detection with Machine Learning

Supervisor: Dr. Kun Yu

School: University of Technology Sydney (UTS)

Table of Contents:

Abstract

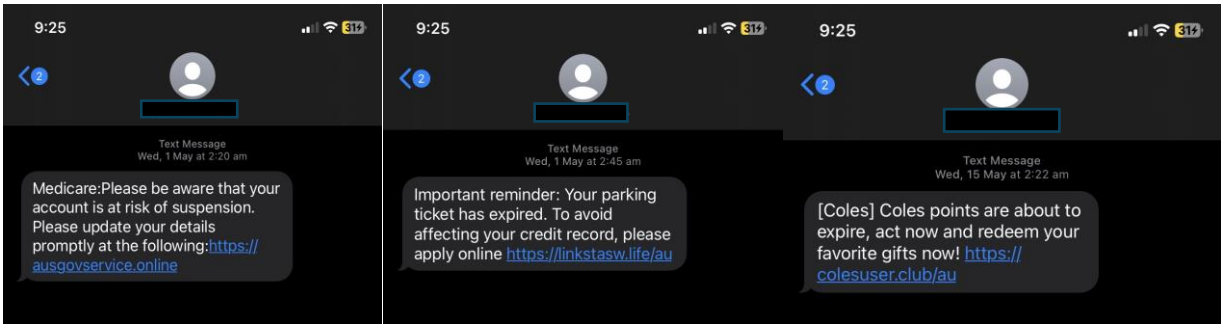
Phishing attacks remain a pervasive cybersecurity threat, leveraging deceptive URLs to exploit human vulnerabilities, steal sensitive information, and compromise systems. Possibly leading to significant financial loss for both individuals, businesses and organisations. This research investigates a fine-tuned machine learning approach for phishing detection using lexical URL features, focusing on optimization techniques such as data balancing, feature selection, and hyperparameter tuning to enhance performance. Two datasets, PhiUSIIL and ISCXURL-2016, were used to Random Forest (RF), Gaussian Naive Bayes (GNB), and Support Vector Machine (SVM) classifiers.

RF achieves a near-perfect accuracy on the recent phishing dataset PhiUSIIL. The feature 'URLSimilarityIndex' was found to be highly discriminative, allowing for a streamlined feature set which exhibited the same or better performance than using all dataset features. While data balancing showed minimal impact on model performance, feature selection and hyperparameter tuning emerged as a significant factors in achieving optimal results.

This study addresses gaps in existing literature by exclusively utilising lexical URL features on a large contemporary dataset, applying optimisation techniques, and proposing a robust, language-independent detection framework suitable for real-time implementation on resource-constrained devices like mobile phones. The research establishes a benchmark for phishing detection with machine learning, in hopes of mitigating identity theft, financial losses, and organizational vulnerabilities.

1. Introduction

Phishing is a simple cyber attack which disguises a malicious website by mimicking a legitimate website. They are commonly sent via text message, phone calls, emails and social media messages, with the sender appearing to be a legitimate authority. This disguise attempts to lull the victim with a sense of trust and entice them into clicking on the malicious link - and entering their details like they would on the legitimate site. Additionally, accessing the malicious link may trigger an automatic download of seemingly legitimate software from the supposed trusted entity. Which, in reality, is malware - which may collect personal information, allow remote access, or even hold files ransom via encryption. Therefore, allowing the attacker to harvest information like login credentials, emails, bank details, credit card details, and demand payment to decrypt files held for ransom.



[Figure 1 – Text Message Phishing Attacks]

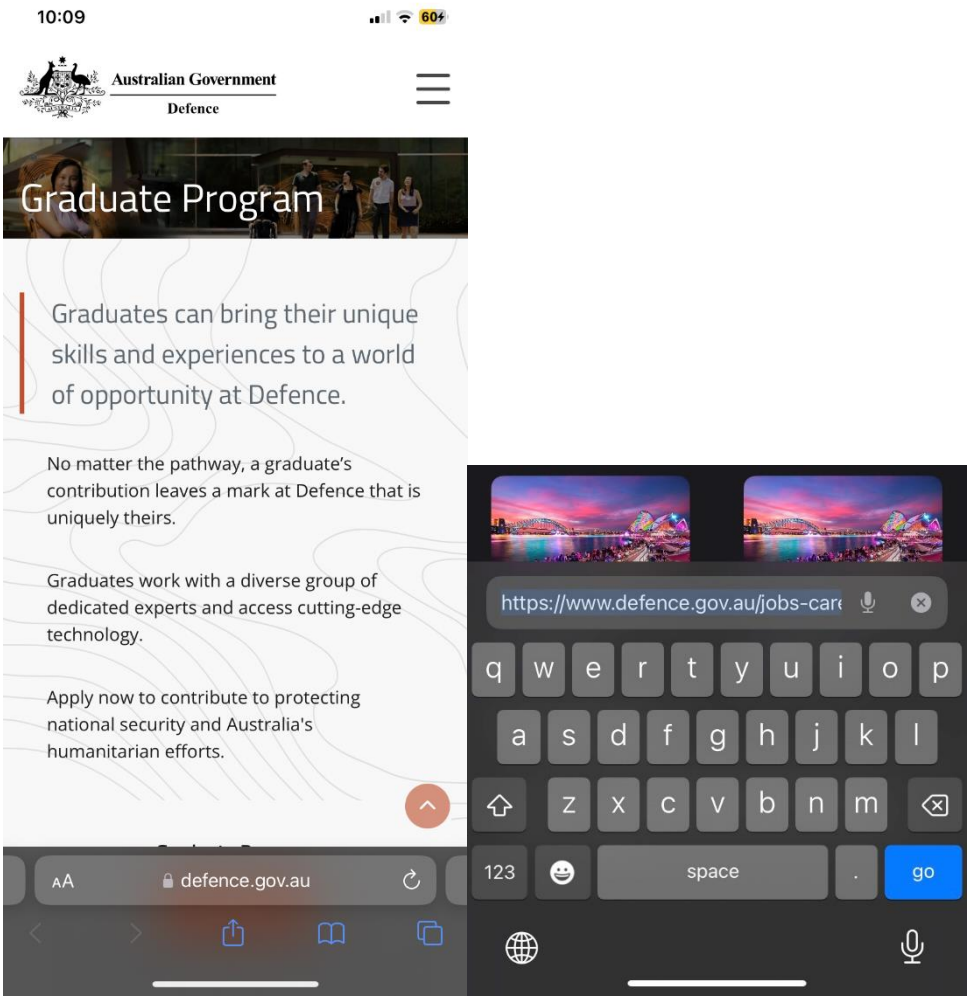
This may possibly lead to identity theft and financial loss, with $84,000 being the average wire transfer amount requested in business email compromise attacks in Q1 2024 – where the attacker poses as a trusted business entity to mislead the employee into clicking a link or sharing confidential information (APWG, 2024). Stolen credentials from phishing can also result in the execution of more dangerous attacks on large businesses. Potentially leading to the violation of the business system's availability, and confidentiality and integrity of customer/employee data. Attackers have moved beyond malware to gain initial access, and towards faster, more effective means like phishing, social engineering, and access brokers with 75% of detections being malware-free activity (CrowdStrike, 2024).

While there were only 963,994 phishing attacks observed by the Anti-Phishing Working Group (APWG) in Q1 2024 (the lowest quarterly total since Q4 2021), 2023 was the worst year for phishing on record, with almost five million phishing attacks observed by APWG. Social media platforms were the most frequently attacked sector – accounting for 37.4% of all phishing attacks in Q1 2024 (APWG, 2024). Thus, indicating that regular individuals are being targeted, not just businesses.

Before the widespread adoption of mobile phones, phishing was typically conducted through computers and laptops as they were the predominant device used personally by individuals. Working in as much the same way as they do now, phishing attempts appeared as emails, social media messages, online gaming messages, and as messages in forums (Do et al., 2022). However, as mobile phones became increasingly popular and their adoption grew, the average individual's attack surface increased with not only existing phishing attempts being able to be conducted on phones, but also via sms messages and phone calls.

Considering personal mobile phones enlarge an individual's attack surface, mobile phones are also becoming increasingly affordable, making it easier for people from all socio-economic statuses to access the Internet - with 5.15 billion people already owning one (Shoaib, 2023). Thus, implying that there are increasingly more people that may be vulnerable to phishing attacks. Especially if they are new phone users and are unaware of how to determine the legitimacy of URLs. In the United States alone, the FBI reported a financial loss of over $1.8 billion in 2020 for the American population (Hillman et al., 2023). Additionally, corporate mobile usage has increased with the implementation of bring-your-own-device (BYOD) policies in response to the shift to working from home during the Covid-19 outbreak (Cinar & Kara, 2023). Thereby, resulting in an enlarged attack surface for businesses employing BYOD.

Mishra and Soni (2019) disclose that there is a lack of awareness amongst users concerning mobile security options. They also stated that this lack of awareness also extends to mobile phone security threats and vulnerabilities, with most people considering their phone more secure than their computer. This view is also shared by Limna et al. (2023) when they disclosed that despite a majority of people having heard of cybersecurity, their sense of urgency and behaviour did not reflect a high level of awareness. Further mentioning that there is strong sentiment regarding the internet as a safe and secure environment for the exchange of information and transactions. Additionally, the display of mobile phones are small. In turn, when a webpage is visited in a mobile browser, the URL is partially hidden as seen in Figure 2. Making it significantly more difficult to try and discern whether the URL is legitimate or not.



[Figure 2 – URL Partially Hidden on iPhone's Safari Web Browser]

According to Verizon's Data Breach Investigations Report (2024), it takes users less than 60 seconds to fall for phishing on average. Furthermore, 68% of all breaches in 2024 involved a non-malicious human element, whereby a person fell to a social engineering attack or made some type of error. While Verizon calls for more training to remedy these poor statistics, Afroz & Greenstadt (2011) indicated that educating people regarding the legitimacy of a website is tedious and time-consuming. Which is not ideal for businesses that operate in a fast-paced work environment. Where employees are preoccupied with and motivated to perform primary tasks for which they are compensated, rather than secondary tasks for which they are not directly compensated (e.g. security training) (Hillman et al., 2023). Thereby, facilitating behaviour within employees to rush through security training modules or even ignore them entirely – and develop a company culture of security policy non-compliance. Ultimately, reducing the effectiveness of training and diminishing the time and money spent on training employees.

Currently, there are solutions on the market to protect against phishing emails, like Microsoft Defender for Office 365 and Norton 360. For mobile devices, there is Norton mobile, McAfee Security and NovoShield. However, Norton mobile does not track iMessage (iPhone's messaging application). Furthermore, all security solutions utilise AI to detect phishing attempts. However, these rely on API calls to the cloud as they are not implemented locally on the device. Thus, if the API goes down, then the device is rendered vulnerable to phishing attempts. Considering it takes users less than 60 seconds to fall victim to phishing, downtime of any duration may be all that is needed for the user to fall victim to phishing.

While there is minimal difference in how a phishing URL works on a desktop computer, versus a mobile phone. There is a difference in how easily the URL can be viewed in browsers. On computer, the entire URL can typically fit in the address bar. Furthermore, more of the URL can be examined simply by highlighting and dragging the URL. However, on a mobile phone browser, the URL is partially hidden, and the action of examining the entire URL by highlighting and dragging is slow. Thereby increasing the user's likelihood of their impatience and laziness to get the better of them and avoid examining the URL for suspicious aspects.

This, in culmination with widespread phone usage, lack of security awareness, how quickly people fall for phishing, the ease of launching a phishing attack, the volume at which phishing is executed, and the damage it can cause - illustrates phishing as a highly dangerous attack. Indicating the need for an accurate solution that responds quickly, and requires no cybersecurity knowledge from the user to detect and defend against phishing attacks, especially on resource constrained devices like phones.

2. Background

Many approaches have been proposed to combat phishing attacks. This includes the likes of blacklists/whitelists, heuristic approaches, and machine learning approaches.

2.1. Blacklist/Whitelist

The most simple and widely used solutions to combat phishing are blacklists and whitelists. They are commonly used by web browsers like Google Chrome, Mozilla Firefox and Safari - which utilise Google Safe Browsing (GSB) as their default protection method. A blacklist contains phishing URLs and a whitelist contains legitimate URLs. If a URL is found to be in the blacklist, access to it triggers a warning, and access is blocked. If a URL is not in the whitelist, an alert is triggered, and access may be blocked. While this approach is highly effective at blocking and allowing access via URLs contained within respective lists, blacklists cannot defend against zero-day phishing URLs (Phishing URLs less than one day old), nor can it detect phishing URLs that are not on the list (Mishra & Soni, 2019). Meanwhile, whitelists can defend against zero-day phishing attacks, as they only allow access to URLs contained within them. However, both whitelists and blacklists fall victim to size constraints - meaning not every single phishing URL can be stored in a blacklist, and not every single legitimate URL can be stored in a whitelist. Therefore, limiting the legitimate websites that can be labeled as trusted and accessed. Which may cause numerous unnecessary alerts to be triggered on legitimate sites, leading to a poor user experience. This also means that phishing URLs within the blacklist must be removed and updated with the most-recent phishing URLs (Do et al., 2022). On top of this, minor changes can be made to a URL to bypass the blacklist (Gupta et al., 2021).

2.2. Heuristic

Heuristic approaches extend upon list-based methods. They utilise various features of web pages to establish predefined rules and patterns to detect the legitimacy of websites. Once evaluated, the URL may then be added to a blacklist or whitelist. Types of heuristic methods include visual similarity, content similarity and URL-based, with all 3 being commonly combined to provide a robust phishing detection approach (Awasthi & Goel, 2022). Visual similarity involves analysing and comparing the visual representation of web pages with images of whitelisted webpages via digital image processing. If the similarity exceeds a certain threshold, then the webpage is considered phishing. However, because of the image processing and storage of images, there is a higher computational requirement than that of URL-based approaches (Do et al., 2022). Content similarity compares webpage content such as keywords, spelling, hidden HTML elements and JavaScript functions. However, Jalil et al. (2023) stated that the visual similarity and content similarity approaches can be overcome by changing a minor portion of the webpage without changing its contents.

Additionally, Basit et al. (2021) also revealed that these approaches exhibited high false positive rates. The URL-based approach involves extracting lexical features from the URL and extracting host-based features from the WHOIS database, like domain registration date, search engine index, and page rank. Jalil et al. (2023) identify that the lexical URL feature-based is the only non-web dependent approach. They also praise its low processing time and the very high URL detection accuracy. Awasthi & Goel (2022) recognise the safety of URL-based similarity in comparison to content similarity, which may process risky features that contain malware – leading to a disruption of the processing system.

2.3. Machine Learning

Machine learning approaches involve training machine learning algorithms to learn complex patterns and relationships in the data. This then allows the model to perform predictions on a given URL concerning its legitimacy. Common machine learning classifiers used in phishing detection include Random Forest (RF), Decision Tree (DT), Naive Bayes (NB), Gradient Boosting (GB), Logistic Regression (LR), and Support Vector Machine (SVM) (Samad et al., 2023; Awasthi & Goel, 2022; Basit et al., 2021; Fajar et al., 2024; Gupta et al., 2021; Jalil et al., 2023; Liu et al., 2023, Thahira & Ansamma, 2022; Vajrobol et al., 2024). They are able to detect obscure patterns and relationships in data that are not apparent via human recognition. Essentially, machine learning approaches automate the rule-creation segment of heuristic methods, but instead of generating rules, they learn complex patterns and relationships in the data. Therefore, allowing it to easily adapt to new phishing attacks when retrained on updated data, whereas heuristic approaches are static and would have to go through the time-consuming process of analysing features and deriving updated rules to combat new phishing attacks. Unlike list-based and heuristic approaches, machine learning approaches are able to generalise very well and detect zero-day attacks.

2.4. Related Works

Phishing URL detection is a well-researched area. However, that does not mean there are gaps or limitations that must be addressed in recent research. as aforementioned, one of the limitations of recent research is

Samad et al. (2023) address one limitation in particular. Specifically, they explore the utilisation of all three optimisation methods out of data balancing (SMOTE upsampling), feature selection, and hyperparameter tuning. Unlike recent research which only focuses on the utilisation of one or two optimisation techniques, rather than the combination of all three.

Experimentation involved machine learning algorithms such as Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), K-Neighbours Classifier (KNN), Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Bernoulli Naive Bayes (BNB), Gradient Boosting (GB), Extreme Gradient Boosting (XGB). They noted minor improvements in performance after data balancing and feature selection. Whereas hyperparameter tuning is attributed to the largest improvement in the performance of models. The most notable performances were that of RF and GB with accuracies of 97.44% and 97.47% respectively on the UCI dataset. Whereas, on the Mendeley dataset GB performed the best with an accuracy of 98.27%. The authors placed emphasis on the fact that these results of the optimised models were better than the previous approaches presented by recent research. The research in this paper is heavily influenced by the work by Samad et al. (2023), particularly the use of data balancing, feature selection, and hyperparameter tuning to optimise the data and models. However, the datasets they used are not representative of contemporary phishing URLs. Furthermore, the datasets used contain a small sample size. The UCI dataset only contains 11,055 samples, and the Mendeley dataset contains 10,000 samples. A dataset with a larger number of samples may provide more reliable and accurate results.

Prasad & Chandra (2023) introduce a Phishing URL detection framework. This consists of a new comprehensive URL phishing dataset in tandem with an incremental learning ensemble made up of BNB, PassiveAggressive (PA), and Stochastic Gradient Descent Classifier (SGD) – which allows the model to continuously learn from new data, rather than retraining the model on the entire dataset again when new data is added. The PhiUSIIL dataset comprises of over 200,000 recent URLs, along with lexical URL features and HTML features. One URL feature in particular, ‘URLSimilarityIndex‘ (USI) helps to detect URL visual similarity-based attacks whereby the attacker creates phishing URLs that look similar to their legitimate counterparts by replacing characters with any that are visually similar. Visual similarity-based attacks include zero-width characters, homograph, punycode, homophone, bit squatting, and combosquatting. USI is a value between 0 and 100 which represents how closely a URL resembles any of the URLs character for character from the top 10 million legitimate websites from Open PageRank Initiative. Where, a value of 100 indicates an exact match to a legitimate URL on the list, and a value close to 100 potentially indicates a phishing URL that looks very similar to a URL on the list. According to the framework, anything with a USI less than 80 is then passed to the machine learning model for further analysis. Their ensemble model achieved an impressive accuracy of 99.24%, and 99.98% with a default RF classifier. Whilst these results are impressive, the approach is computationally expensive (3 algorithms) when compared to the use of a single algorithm for classification. The authors also acknowledge that overfitting is a limitation of their continuous learning approach. Furthermore, it also relies on the utilisation of HTML features. Which, as mentioned above, requires visiting the URL to extract in a practical scenario. Thereby leaving the user potentially vulnerable to the automatic execution of malicious scripts when loading up the web page. Additionally, only hyperparameter tuning was performed on the ensemble model, with data balancing and feature selection being absent.

Furthermore, the default RF model performed better than the ensemble model, further suggesting the possibility of achieving better performance via hyperparameter tuning.

Vajrobol et al. (2024) achieved a performance of 99.97% with their Logistic Regression (LR) model. Their approach was centered around combining mutual information (MI) feature selection with logistic regression to more accurately detect phishing URLs by using a more optimised version of the PhiUSIIL dataset that was merely cut down to 5 features, 'URLSimilarityIndex', 'LineofCode', 'NoOfExternalRef', 'NoOfImage', and 'NoOfSelfRef'. The authors also reported a decrease in accuracy, precision, recall, and f1 as the number of features increased. They, suggested that larger feature sets may increase model complexity. Which, in turn may affect the efficacy of the model. One issue with this approach is that it leaves the user vulnerable due to the use of HTML features. It is further limited, in that only feature selection was utilised to optimise the approach. There may be room for the approach to achieve improved performance with the additional incorporation of data balancing and hyperparameter tuning.

DomURLs_BERT achieved a performance of 99.80% on the PhiUSIIL dataset (Mahdaouy et al., 2024). Their deep-learning approach using Bidirectional Encoder Representations from Transformers is pre-trained on a large-scale multilingual corpus of URLs, domain names, and Domain Generation Algorithms (DGA) datasets. Extensive valuation was performed on 11 different datasets, and binary and multi-class classification tasks. It is able to successfully classify DGA, DNS tunneling, malware, and phishing tasks. As a result, it outperformed 6 state of the art character-based deep-learning models and 4 BERT-based models. The datasets also contained recent data as a majority of the datasets were created in 2024, with the oldest being from 2019. Thereby, providing a comprehensive illustration of the performance of their approach on recent phishing/malware URLs, and DGA/DNS tunneling domain names. One limitation of this work, however, is that no performance comparisons are made to state of the art traditional machine learning approaches. This would provide a clearer idea of how the approaches measure up against each other, but also support the authors' argument for pursuing deep learning over traditional machine learning. That being, machine learning involves manual feature engineering, which is costly and time-consuming.

Fajar et al. (2024) evaluated RF, CatBoost, XGB, and Explainable Boosting Machine (EBM) on the PhiUSIIL dataset. RF, CatBoost, and EBM were able to achieve 100% accuracy. The main focus of this study was feature selection. As such, they found that 'URLSimilarityIndex' was very decisively the most dominant feature in the dataset out of the 54 features in PhiUSIIL. So much so, that with XGB, it was able to achieve an accuracy of 99.6% using 'URLSimilarityIndex' alone. Whereas RF was able to achieve its remarkable accuracy on 12 features, CatBoost on 4, and EBM on 17. In addition to feature selection, upsampling using SMOTE was also performed to balance the PhiUSIIL dataset. However, the quality of the synthetically generated samples was never verified or validated. Furthermore, no hyperparameter tuning was performed despite the models also being evaluated on 8 other datasets.

Also, both the URL and HTML features from the dataset were used. Results are inconsistently portrayed, in that percentages are sometimes rounded to the nearest whole number or the first decimal place. Considering the remarkable performance of models in the PhiUSIIL dataset in not only this paper, but the previously mentioned. The results would be far more reliable if rounding was consistent to atleast two decimal places to provide a more granular representation of performance. For it is unclear whether a 100% accuracy was actually achieved, or if it was rounded up from a score such as 99.97%. The authors do not explicitly state their method of rounding.

Muntean (2024) evaluated their PART Decision Rules Classifier (DR) on the PhiUSIIL dataset using URL and HTML features and reported an accuracy of 99.99%. They utilised this model from the Weka Machine Learning Tool library. The author focused on documenting their data preprocessing, particularly utilising the String to Nominal filter (similar to label encoding) on categorical data, and the Discretize filter (similar to ordinal binning) on the numerical features 'URLLength' and 'DomainLength'. This research is limited by its use of HTML features. It also does not apply any data balancing, feature selection, or hyperparameter tuning.

By integrating Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory networks (BiLSTM) into an ensemble, the authors have achieved an accuracy of 89.5% on the PhiUSIIL dataset with their NLP approach (Firdaus & Sumardi, 2024). This was achieved by converting the URLs into numerical representation via TF-IDF Vectorizer, which is then fed into the CNN for feature extraction, followed by BiLSTM to capture long-term dependencies in textual data. It is not specifically mentioned whether the original features were retained and utillised or not. However, it is assumed that only the URL strings were utilised from the dataset. The authors argument for utilising a deep-learning ensemble stemmed from the biases and weaknesses associated with using single-model approaches. Claiming that ensemble methods lead to improved detection accuracy as they combine the strength of multiple models. While this would generally stand true for a model such as RF, the performance of their approach spercifically does not coincide with this claim when compared to the performances of other recent approaches evaluated on the PhiUSIIL dataset. Especially the performances of single-model approaches. Furthermore, the performance of the approach was not compared to any other existing approaches on the same or different datasets. Thereby making it difficult to gauge whether the accuracy is relatively performant within the content of the specific method they are pursuing for their approach.

Using only 9 lexical URL features, the authors were able to achieve an accuracy of 99.57% on their RF model in a real-time environment (Gupta et al., 2021). Furthermore, SVM also achieved an accuracy of 97.64%. These features are "No. Of token in domain", "No. Of top level domain", "Length of URL", "No. Of dots in URL", "No. Of delimiter in domain", "No. Of delimiter in path", "Length of longest token in path", "No. Of digit in a Query", and "Domain length". The authors also found that "No. Of delimiter in path", "No. Of delimiter in domain", "No. Of dots in URL", and "Length of URL" are the most important features out of the 9.

This work is limited by its use of the ISCXURL-2016 dataset for training and evaluation. It does not contain recent phishing URLs, and it also has a small number of samples, with only 19,964 samples.

Thahira & Ansamma (2022) acknowledged that a majority of existing solutions were ineffective for resource-constrained devices like mobile phones. Hence, their lightweight solution utilising lexical features to achieve an accuracy of 99.5% on the ISCXURL-2016 dataset using the Light Gradient Boosting Machine (LGBM) from the FLAML library (Fast and Lightweight AutoML). However, this performance on the ISCXURL-2016 can be estimated using their graph. As previously mentioned, valuations were also performed on two additional datasets, with the third being a dataset created by the authors using Phishtank and Openphish. However, both of these datasets have insufficient numbers of samples. Dataset 2 only contains 11,430 URLs, and the third dataset has 20,630 URLs. It should also be noted that only the performance on the ISCXURL-2016 dataset is included for comparison, as the other datasets used by the authors were not evaluated on within this paper. Additionally, the only optimisation techniques used were feature selection and hyperparameter tuning.

Rugangazi & Okeyo (2023) achieve an accuracy of 98.66% on the ISCXURL-2016 dataset with their RF model – which outperformed LR and KNN. Their approach was centered around the comparison of feature selection methods - forward regression and feature importance. Interestingly, the evaluation using all features outperformed the other evaluations utilising either feature selection method for every model. For RF, forward regression achieved 98.57%, and feature importance garnered 98.59%. Despite this, the authors still insisted that using either feature selection method would still be beneficial despite the minor drop in performance. This was the case as the tradeoff for lower computational requirements was deemed far more valuable. The performance of the models may also have been improved if data balancing or hyperparameter tuning was conducted. Oddly, the authors did not disclose the features in particular that were selected for either feature selection method. This would have been beneficial for comparison. To extend on this, it would also have been further beneficial for comparison if another dataset were used. Preferably, a more recent dataset with a large number of instances.

Liu et al. (2023) approached the ISCXURL-2016 dataset differently in that instead of utilising the 10,000 legitimate URLs from the phishing subset, they used all 35,378 legitimate instances from the entire dataset. This imbalance gave the authors the means to explore a more realistic evaluation of the performance of their approach. One which similarly reflects the imbalance of benign and phishing URLs in real life. Hence, giving a strong argument in support of not performing data balancing, and pursuing a realistic evaluation over a maximised evaluation of performance. Furthermore, they established their own feature set, made up of Compositional features and Linguistic features – totalling to 33 features. Evaluations were performed in real-time on RF, XBG, DT, KNN, SVM, LR, NB, Artificial Neural Network (ANN), and Gradient-Boosted Decision Tree (GDBT). RF performed the best with an accuracy of 98.8% on all 33 features.

As such, RF was chosen as the main classifier for their phishing detection framework, which was fleshed out with a component analysis, text decomposition, and readability check modules before feature extraction of the given URL. Two other datasets were also used for evaluation, with the first being of the authors' own creation. With recency of URLs in mind, they used 209,445 legitimate URLs from Tranco, and 44,049 from PhishTank. The third dataset was derived from Sahingoz et al.'s (2019) which contained 36,400 legitimate URLs, and 37,175 phishing URLs. With these datasets, Liu et al. were able to provide a comprehensive overview of the performance of machine learning models on old and recent unbalanced datasets, balanced datasets, and datasets with small and large numbers of instances. There was no mention of hyperparameter tuning having been performed.

Ref.	Features	Model	Dataset	Accuracy	Limitations
Samad et al. (2023)	URL, HTML, Domain	-	UCI, Mendeley	-	Old URLs, small sample size.
Prasad & Chandra (2023)	URL and HTML	Ensemble (BNB, PA, SGDC), RF	PhiUSIIL	0.9924 (Ensemble), 0.9998 (RF)	Utilises HTML features, relatively computationally expensive model (ensemble – 3 algorithms), continuous learning may risk overfitting, only hyperparameter tuning used.
Vajrobol et al. (2024)	URL and HTML	LR	PhiUSIIL	0.9997	Utilises HTML features, only feature selection used.
Mahdaouy et al. (2024)	URL and HMTL	DomURLs_ BERT	PhiUSIIL	0.9980	Utilises HTML features, computationally expensive model (deep-learning), no comparisons made to traditional machine learning approaches.
Fajar et al. (2024)	URL and HTML	RF, CatBoost, EBM, XGB	PhiUSIIL	1 (RF, CatBoost, EBM), 0.996 (XGB)	No hyperparameter tuning, SMOTE sample quality is not verified, accuracy rounding is not granular enough and could be inflating scores.
Muntean (2024)	URL and HTML	PART-DR	PhiUSIIL	0.9999	Utilises HTML features, no data

					balancing, feature selection, or hyperparameter tuning.
Firdaus & Sumardi (2024)	URL and HTML	Ensemble (CNN-BiLSTM)	PhiUSIIL	0.895	Computationally expensive model (deep-learning ensemble), no comparisons made to any existing approaches, relatively poor performance.
Gupta et al. (2021)	URL	RF, SVM	ISCXURL-2016	0.9957 (RF), 0.9764 (SVM)	Old URLs, small sample size, no data balancing (Didn't disclose if pursuing realistic evaluation)
Thahira & Ansamma (2022)	URL	LGBM	ISCXURL-2016	0.995 (Estimated from graph)	Doesn't explicitly disclose performance of LGBM on ISCXURL-2016 dataset, old URLs, small sample size.
Rugangazi & Okeyo (2023)	URL	RF	ISCXURL-2016	0.9866	Old URLs, small sample size. No data balancing or hyperparameter tuning, does not disclose selected features from forward regression, nor feature importance.
Liu et al. (2023)	URL	RF	ISCXURL-2016 (Used all 35,378 URLs)	0.998	No hyperparameter tuning, no data balancing (pursuit of realistic evaluation).

[Table 1 – Summary of Related Work (Model and Accuracy comparisons only for PhiUSIIL and ISCXURL-2016 datasets)]

2.5. Research Gaps

Gupta et al. (2021) found that a number of machine learning approaches suffer from high response times (the time between the URL fed, and the result predicted) due to dependency on third-party features like whois records, DNS records, and blacklist databases. This also introduces availability issues in the case of downtime or the discontinuation of their service. Discontinuation also raises reliability issues as the features would no longer be updated.

Furthermore, approaches which utilise HTML and/or webpage-visual similarity features, in a practical setting, requires visiting the URL to extract these features. Thus, leaving the user susceptible to the automatic execution of malicious scripts on visit to the phishing URL (Jalil et al., 2023). Similarly, Awasthi & Goel (2022) recognise that processing HTML content involve processing risky features that contain malware – leading to a disruption of the processing system.

To that end, lexical URL features are commonly utilised to remedy these issues (Gupta et al., 2021; Jalil et al., 2023; Thahira & Ansamma, 2022).

Samad et al. (2023) stated that most of the present literature surrounding phishing detection using machine learning focuses on enhancing the models by proposing novel aspects. Meanwhile, less attention is paid to the fine-tuning factors like data balancing, feature selection, and hyperparameter tuning. This statement is seems to be true considering the lack of optimisation techniques being utilised within recent works (Prasad & Chandra, 2023; Vajrobol et al, 2024; Fajar et al, 2024; Muntean, 2024; Gupta et al., 2021; Rugangazi & Okeyo, 2023; Liu et al., 2023). With approaches only using one or two of the fine-tuning techniques, rather than the utilisation of all three. However, in some cases unbalanced data sets are purposefully utilised to mimic real life conditions and provide realistic performance evaluations (Liu et al., 2023).

Finally, recent approaches are evaluated on old datasets (Gupta et al., 2021; Jalil et al., 2023; Thahira & Ansamma, 2022), which do not provide a true representation of these approaches' performances on the latest phishing URLs. Another issue is also the number of samples within the dataset. Many recent approaches are only evaluated on small datasets (Samad et al., 2023; Gupta et al., 2021; Thahira & Ansamma, 2022). A dataset with a higher number of samples may be able to facilitate a model's discovery of hidden relationships, and thus, improve performance. Furthermore, a higher number of samples also enhances the reliability of results.

2.6. Research Aims and Objectives

To effectively combat the evolving landscape of phishing attacks and mitigate the consequences associated with a lack of security awareness, this paper proposes a fine-tuned machine learning solution that uses lexical features from URLs to accurately detect phishing URLs. The following are the research aims and objectives set out at the time of this research being proposed.

- Aim 1: To investigate the performance of various Machine Learning Algorithms using lexical URL features for the detection of the latest phishing URLs.
    - Objective: I will train and test the performance of Random Forest, Gaussian Naive Bayes, and Support Vector Machine classifiers on the PhiUSIIL dataset.
    - Success Criteria: The illustration of each classifier's accuracy, recall, precision and F1 score in tables and graphs.

- Aim 2: Fine-tune classifiers using lexical URL features to outperform existing approaches that do or do not use lexical URL features for phishing detection.
    - Objective: Perform data balancing, feature selection, and/or hyperparameter tuning on classifiers.
    - Success Criteria: At least one of the fine-tuned models perform better than one of the existing approaches that do or do not use lexical URL features.

- Aim 3: To verify if data balancing improves performance.
    - Objective: Upsample data labels so that dataset is balanced.
    - Success Criteria: Performance evaluation post-data balancing indicates improved accuracy, recall, precision and/or F1.

- Aim 4: To find the minimum amount of features needed to achieve the same or better performance than that of using all features in the dataset.
    - Objective: Perform feature selection for Random Forest, Naive Bayes, and Gaussian Naive Bayes classifiers.
    - Success Criteria: Selected feature subset contains less features than the number of total features, and achieves the same or better performance than using all features.

- Aim 5: To verify if hyperparameter tuning improves performance.
    - Objective: Perform hyperparameter tuning on Random Forest, Naive Bayes, and Gaussian Naive Bayes classifiers.
    - Success Criteria: Performance evaluation performed post-hyperparameter tuning indicates tuned classifiers (atleast one hyperparameter is not default) achieved improved accuracy, recall, precision and/or F1.

2.7. Research Rationale

Phishing attacks are one of the easiest cyber attacks to launch, due to the relatively low-technical knowledge and expertise required to launch them. It can impact the everyday person with their details being stolen and login details being compromised. On the other hand, it can cost large businesses millions of dollars if an employee's login credentials are compromised, and are utilised to facilitate further devastating attacks.

Any device that can connect to the Internet is vulnerable, especially devices that are commonly used every day like computers, laptops, and phones, and smartwatches. However, moreso vulnerable are resource-constrained devices that have small displays, like mobile devices and smart watches. Consequently, it makes it difficult to view the entire URL and discern if it is malicious or not.

Phishing preys upon human behaviour. Due to this, it is tedious and time consuming to train people on how to detect phishing websites and defend themselves. Thus, a more cost-effective and immediate solution is required to detect and prevent phishing attempts. Especially one which requires no cybersecurity knowledge from the user.

2.8. Future Impact

This project aims to contribute to the academic understanding and practical execution of phishing detection using lexical URL features by emphasizing the combined optimisation of data balancing, feature selection, and hyperparameter tuning to showcase the true potential of proposed solutions. Additionally, this is likewise regarding the use of more recent datasets to demonstrate the true performance of proposed solutions on the latest phishing URLs. It is hoped that this provides a template and sets the standard for future research and development within this field.

Furthermore, this paper also makes contributions in regards to optimal selection of features and hyperparameters for the recent PhiUSIIL phishing URL dataset by Prasad et al. (2023).

Future continuation of this work would involve establishing a phishing detection framework involving feature extraction and each of the models in this work. Then, evaluating the performance of the framework in real time, and recording the processing time. Following this research would then be a local mobile phone integration of the framework and again performing a real time evaluation, also including processing time to find which model requires the least computational resources.

It is hopeful a local implementation of the framework would circumvent the need to utilise API calls to a remote model whose availability may be impeded if the server goes down unexpectedly, or is subject to availability attacks like a Distributed Denial of Service (DDoS) attack. Whereby, an attacker floods the server with internet traffic from multiple sources. Preventing it from processing packets from legitimate users. Ultimately, causing the user to resort to discerning the legitimacy of URLs themselves in the absence of the phishing detection model.

I am hopeful that in the future, this model would be implemented through an app available on the iOS App Store and Google Play for Android devices, to which whenever a URL is accessed within the phone's web browser, it is checked and classified by the phishing detection model. If it is classified as phishing, an alert is shown and access to the webpage is blocked.

Consequently, mitigating the risk of falling victim to a phishing attack due to the lack of security awareness exhibited by phone users (Mishra & Soni, 2019) and the low priority given to security training by employees in large companies (Hillman et al., 2023). This is mainly due to security knowledge not being a requirement for the phishing detection model to detect phishing URLs. Nor does it impose additional burdens or require extensive training, which allows employees to fully focus on their primary task for which they are compensated for. Therefore, also reducing the attack surface of companies, especially those that implement BYOD. This could then have a trickle effect and reduce the volume of major attacks on businesses. Particularly, attacks which commonly rely on phishing as a preliminary step in gaining access to business systems by attempting to trick employees into entering their login credentials.

Due to the utilisation of lexical URL features, phishing URL detection is language-independent and is not merely limited to English webpages. Furthermore, being a machine learning classifier, the solution would be robust against evolving phishing attacks and zero-day attacks, provided the classifier is retrained on the latest phishing URLs when appropriate.

However, the most anticipated future impact would be the reduction in identity theft, and financial loss for both individuals and businesses. Thereby fostering a safer and more secure internet environment for users to enjoy.

The rest of this paper is organised as follows: Section 3: Method, Section 4: Results, Section 5: Discussion, and Section 6: Conclusion.

3. Method

3.1. Dataset

The first dataset used for the training and testing of the classifiers is the PhiUSIL dataset (Prasad & Chandra, 2023) from the UCI archive. It is a comprehensive collection of data which emphasises the integration of the latest phishing and benign URLs. It contains 134,850 legitimate and 100,945 phishing URLs. It also contains 54 features extracted from the webpage and URL. However, for the sake of this research, only the URL features are utilised. Thus, after HTML features were removed and extracted features were added, the number of features used amounted to 30. An 80:20 train/test ratio is used. Original target classes for the dataset corresponded to 0 as phishing and 1 as benign. However, to make the implementation easier, these were switched so that 1 corresponded to phishing, and 0 to benign.

The second dataset used to train and test the classifiers is ISCXURL-2016 (Mamun et al., 2016), from the data repository of the University of Canada Brunswick. Despite being an old dataset, it provided insight in how this work's approach compared to existing lexical URL feature-based approaches. It contains spam records, phishing URLs, benign URLs, malware and defacement. However, considering the scope of this research, only the phishing and benign URL subset will be used. There are 7,781 benign URLs and 7,586 phishing URLs. It is made up of 79 lexical features extracted from URL, domain, path, file-name, and argument. An 80:20 train/test ratio will be used. Target classes were encoded so that 1 corresponded to phishing, and 0 to benign.

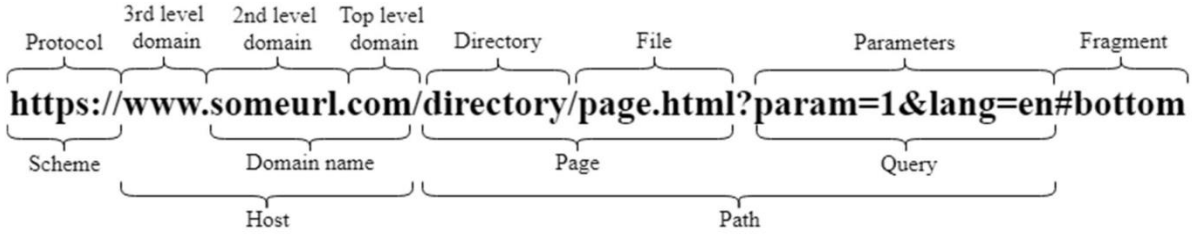
Dataset	Number of Samples	Number of Benign and Phishing Samples	Number of Features	Types of Features	Target Class
PhiUSIIL (Prasad & Chandra, 2023)	235,795	Benign: 134,850 Phishing: 100,945	54 (30 used after removing HTML features and adding extracted features)	URL	1: Phishing 0: Benign
ISCXURL-2016 (Mamun et al., 2016)	15,367	Benign: 7,781 Phishing: 7,586	79 (78 after preprocessing)	URL	1: Phishing 0: Benign

[Table 2 – Datasets used for Training/Evaluation]

3.2. URL Characteristics

To identify a webpage, the Universal Resource Locator (URL) is used (Gupta et al., 2021). The structure of a URL is illustrated in diagram in Figure 3, made by Jalil et al. (2023). The main components of a URL are:

- **Scheme/Protocol:** The protocol of the URL. This specifies how communication between the browser and the server should take place.
- **Host:** A unique reference which allows the website to be identified. It is made up of the subdomain, main domain, and top level domain (TLD).
- **Path:** The address of where the specific resource is located in the web server.
- **Query:** The key-value pairs of data succeeding the '?'.
- **Fragment:** Starting with a '#', it identifies a specific section or element in the web page.



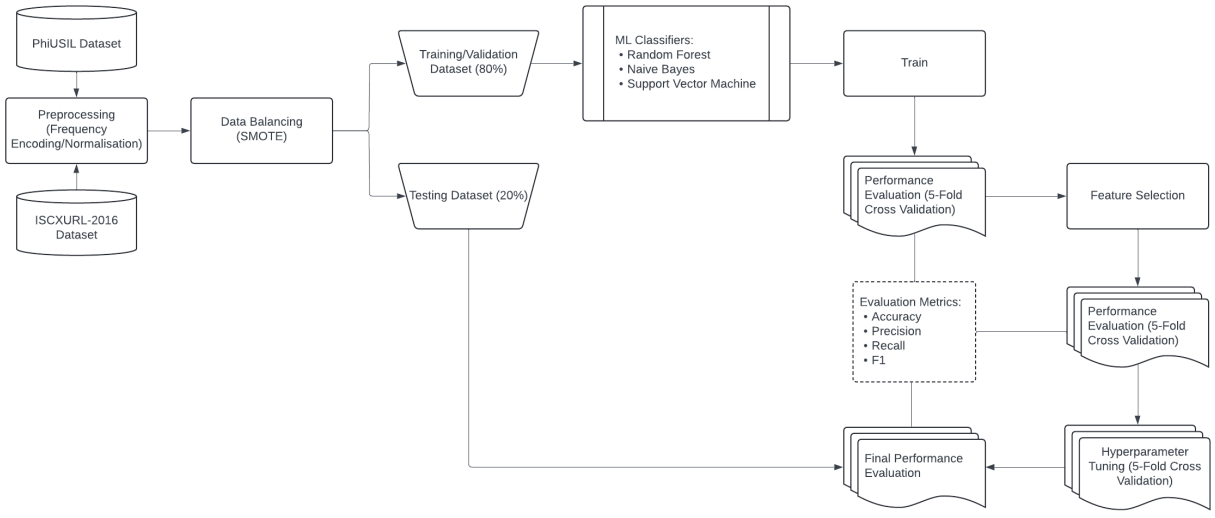
[Figure 3 - URL Anatomy (Jalil et al., 2023)]

3.3. Research Methods

The PhiUSIIL and the ISCXURL-2016 datasets both execute the same process flow as pictured in Figure 4.

Note: Preprocessing and upsampling occur in a pipeline and are performed every time models are evaluated. However, they are represented as such in Figure 4 for the simplicity and neatness of the diagram.

1. The dataset is preprocessed. Categorical features are frequency encoded, and numerical features are normalised using standard scaler.
2. Upsampling is performed on the dataset so that classes are balanced.
3. The dataset is split into a training set and a testing set.
4. RF, GNB, and SVM are evaluated via 5-fold cross-validation (CV) on the training set when balanced and imbalanced. Whichever variation of the dataset performs the best is used for further analysis.
5. Feature selection is performed, followed by post-feature selection evaluation using 5-fold CV. If the number of selected features is less than the original number, and the same performance or better is observed than that of the performance on the original number of features – then the selected number of features is used for further analysis.
6. Hyperparameter tuning is performed on each of the models.
7. Final evaluations of the optimised models are performed on the test set.



[Figure 4 - Process Flow for the proposed approach]

3.3.1. Data Preprocessing

For the PhiUSIIL dataset, all HTML features columns were removed. Then, extracted feature columns were added. Target label encoding was also performed so that phishing corresponded to 1, and benign to 0. The dataset contained no NaN or infinite values. All categorical features except for 'TLD' were not relevant, and were therefore removed.

On the other hand, ISCXURL-2016 is purely made up of URL features. Thus, no features were removed. Originally, target classes were 'benign' and 'phishing'. These were then encoded so that 1 corresponded to phishing, and 0 corresponded to benign. Nan values in 'avgpathtokenlen', 'NumberRate_DirectoryName', and 'NumberRate_FileName' were filled with the value 0. NaN and infinite values in 'NumberRate_AfterPath', 'Entropy_DirectoryName', 'Entropy_Extension', 'Entropy_Afterpath', and 'ArgPathRatio' were filled with the value –1. 'NumberRate_Extension' was completely removed due to just less than half of the feature values being NaN values.

Categorical features in PhiUSIIL were transformed into numerical values using frequency encoding. This simply replaced each category with its frequency in the dataset. This was mainly due to how the only categorical feature 'TLD' was nominal and had high cardinality.

Numerical features in both datasets have diverse data ranges. Hence, the application of standard scaling which transforms data to have a mean of 0 and a standard deviation of 1. Thereby, centering and scaling the date. This fostered uniformity across the dataset and ensured all features were compared fairly despite their original scales.

$$z = \frac{x - \mu}{\sigma}$$

z = Scaled Value

x = Original Value

μ = Mean of feature

σ = Standard Deviation of feature

3.3.2. Feature Engineering

I utilised the 9 lexical features implemented by Gupta et al. (2021) from the feature selection they performed on the ISCXURL-2016 dataset as seen in Table 3. This is mainly due to the impressive performance of their model in terms of it's 99.57% accuracy using only the 9 features, and low response time involving classification within a millisecond.

The PhiUSIIL dataset contained three of the features exhibited in Table 3, namely 'No. Of top level domain', 'Length of a URL', and 'Domain length'. The other six features that were't already included in the dataset were extracted from the URLs using Python scripts utilising the urllib.parse library. Extracted features were then used in tandem with the features already in the dataset.

No.	Feature Name
1	No. Of token in domain
2	No. Of top level domain
3	Length of a URL
4	No. Of dots in URL
5	No. Of delimiter in domain
6	No. Of delimiter in path
7	Length of longest token in path
8	No. Of digit in a Query
9	Domain length

[Table 3 – The Lexical URL Features Used by Gupta et al. (2021)]

3.3.3. Data Balancing

Because both datasets are uneven with legitimate URLs being the majority class, phishing samples are upsampled to match the number of respective legitimate URLs in both datasets using Synthetic Minority Oversampling Technique (SMOTE). This was done in order to attempt prevent results that are skewed in favour of the majority class, in this case, results would be skewed in favour of legitimate URLs for both datasets. Meaning, even if a model were to get a high accuracy, this could be largely attributed to the model correctly classifying legitimate URLs, rather than it correctly classifying phishing URLs. Additionally, oversampling is more ideal than removing data via undersampling as it could result in the loss of critical features (Samad et al., 2023). The peformances of models on unbalanced and balanced datasets were also compared, with the dataset that performed the best being chosen as the main dataset to be used for further analysis.

3.3.4. Model Evaluation

The performance of RF, GNB, and SVM classifiers on both datasets will be evaluated using the performance metrics: accuracy, precision, recall, and F1 score using an 80:20 train/test split on both datasets. Additionally, a confusion matrix accompanies each evaluation to detail the number of FP and FN.

Evaluation on the training set involved the utilisation of 5-fold CV, where each score is presented by its mean of the 5 folds.

Further analysis on the performance of RF on the dataset was conducted to verify the excellent performance of the model at baseline evaluation. This involved evaluations on subsets of 1000 and 100 of the training set. This time using 10-fold CV.

Final evaluations of the optimised models involved a pseudo 5-fold CV approach. Where, the training and testing sets were randomly split into 5 equal subsets each. Each optimised model was then trained and tested on each of the 5 subsets. Each score is presented by its mean of the 5 folds. The same evaluation process was also conducted on each default model to provide a point of comparison for their corresponding optimised counterparts.

1. **Accuracy:** The correctly predicted number of phishing and legitimate URLs out of all samples in the dataset.

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

2. **Precision:** The total number of identified phishing URLs that are actually phishing URLs.

$$\text{Pr}\,e\,cision = \frac{TP}{TP\ +\ FP}$$

3. **Recall:** The total number of correctly identified phishing URLs out of all phishing URLs.

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

4. **F1:** The harmonic mean of precision and recall. Indicates overall performance and optimisation of the anomaly detection model.

$$F1 = 2 \times \frac{\text{Pr}\,e\,cision \times Recall}{\text{Pr}\,e\,cision + Recall}$$

		Prediction	
		Phishing (1)	**Legitimate(0)**
True Label	**Phishing (1)**	TP	FN
	Legitimate (0)	FP	TN

[Figure 3 - Confusion matrix layout]

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative.

3.3.5. Feature Selection

All classifiers have undergone feature selection to explore the features it considers most important. Multiple methods of feature selection were employed:

- RF: Gini Importance/Mean Decrease in Impurity (MDI)
- GNB: Sequential Feature Selection – Forward Selection (SFS)
- SVM: Mutual Information (MI)

MI was utilised as a more general approach to feature selection. Whereas, MDI and SFS were used as more tailored methods. SFS was to be used for SVM as well. However, due to hardware limitations, the time to execute was exorbitant and MI was used instead. MI was also utilised to verify the exceptional performance of RF on PhiUSIIL, when the feature distribution of the top 7 features with an MI >0.2 in PhiUSIIL was examined.

MI calculates the statistical dependence of each feature in regard to the target variable. Meaning, just by knowing the values of various features, it can assess the amount of information learned about the target variable.

MDI identifies the most important features by examining how effectively they reduce impurity in decision splits. With impurity being the likelihood of incorrect classification at a node. Their effectiveness is judged by their Gini Importance score.

SFS operates in forward selection by starting with no features. Then adding features one by one based on their contribution to improving performance. This continues until adding more features ceases to significantly improve performance.

Features that were not considered important were removed. The minimum number of features needed to exhibit performance that was same or better than the performance of each model on all features in the dataset was also found.

3.3.6. Hyperparameter Tuning

Hyperparameters are the parameters that control the learning process of the machine learning model. Finding the most optimal hyperparameters can result in significant increases in model performance, faster convergence, reduced overfitting, and efficient use of computational resources (Saez-de-Camara et al., 2023). F1 was the chosen metric for hyperparameter tuning. Validation curves were plotted to explore a general range of optimal values to be used for GridSearchCV. This tests every single variation of given hyperparameter values in a brute-force manner.

For SVM hyperparameter tuning, it was unreasonable to perform GridSearchCV in some cases due to the time to execute. Same with the use of RandomSearchCV. Therefore, manual hyperparameter tuning was performed using validation curves, and GridSearchCV was employed in some cases where there were a very low number of variations to be tested.

3.3.7. Shapley Additive Explanations (SHAP)

After baseline evaluations were conducted, each model exhibited extremely high performances on PhiUSIIL. As RF was the best performing model, it was used to conduct further analysis to verify the given results and explore why they were exhibited. This involved calculating the SHAP values for each feature in PhiUSIIL, which illustrated the contributions of each feature in predicting whether a sample was phishing or benign.

3.4. Experimental Design

Experiments were performed on a Windows 10 machine with an Intel Core i5-13600K 3.50GHz processor, 16GB RAM, and an NVIDIA GeForce RTX 3080. Experiments were conducted using Python 3.12.2 within a Jupyter Notebook environment. All the machine learning algorithms were from the scikit-learn library, along with methods of evaluation, feature selection, data preprocessing transformers, and calculation of SHAP values. Pandas was utilised for data manipulation, particularly for data preprocessing, and matplotlib was used to visualise data and results on graphs for better understanding. For feature extraction, urllib.parse was used to tokenise the URLs in PhiUSIIL.

3.5. Reliability/Validity Discussion

The reliability and reproducibility of evaluations was ensured through the use of random_state.

Validity was ensured via the utilisation of CV. Which gave an accurate output of results. Therefore, providing meaningful data that is trustworthy and actionable. It also ensures reliability by testing the model on different folds. To which the mean score of all folds is then taken to reflect a more reliable evaluation of the model's performance.

Furthermore, validity was also provided through the training and testing of classifiers on the PhiUSIL dataset. Consequently, this provided an accurate representation of the each of the models' performance on the latest phishing and legitimate URLs.

Validity was meant to also be achieved through data balancing, specifically oversampling rather than undersampling to balance the number of samples in each class and prevent bias towards predicting legitimate URLs, considering it is the majority class in both datasets. However, as evaluations on balanced data exhibited worse performance than that of unbalanced, unbalanced data was used for subsequent evaluations. This then provided validity in a different manner, whereby it gave a truthful representation of each models' performance under conditions which were more representative of real life (less phishing URLs than benign).

Finally, validity was ensured through further analysis of the baseline RF evaluation results on PhiUSIIL to find why such performative results were observed. This consisted of testing the default RF model on subsets of size 1000 and 100 from the training set. Along with examinations of one of the DTs used in the RF model, the feature distribution of the most important features via MI feature selection, and calculation of the SHAP values for each feature. Which then lead to additional evaluations being conducted on PhiUSIIL with the removal of the 'URLSimilarityIndex' feature to provide insight into the performance of the dataset without the use of such a polarising feature.

3.6. Limitations

One of the limitations of this research is the hardware in which this research was conducted on. This resulted in less than ideal solutions to be used, such as 5-fold CV rather than 10-fold CV for all evaluations as 10-fold took an unreasonable amount of time to execute. Furthermore, hardware also impacted SVM hyperparameter tuning in that GridSearchCV and RandomisedSearchCV could not be used properly to test combinations of hyperparameter values as it would taken far too long to execute. This was also observed when trying to perform SFS on SVM. Which then led to MI being used for feature selection for SVM instead.

Additionally, there may be inherent bias in the datasets. For instance, perhaps a majority of phishing URLs were gathered from a particular region, whereby their phishing attacks can be distinguished from those launched from other areas in the world.

Therefore, providing a misrepresentation of the models' performance on phishing attacks in general as the models may perform well one dataset - but then perform poorly on another dataset, where phishing URLs were gathered from a different region.

3.7. Ethical Stance

The datasets used are publicly available on moderated repositories, which would flag and remove any datasets that breach their code of ethics. Despite this, some phishing URLs may include an IP address and have a domain that reveals a geographic region such as ".au" or ".uk". However, these are not enough to try and discern the identity of specific individuals. Especially considering that even though the latest URL are within PhiUSIL, they are still a year old from the time of writing this report.

This research is ethical and aims to improve people's lives by protecting them from potential dangers like identity theft and financial loss after being exploited by a phishing attack.

4. Results

This section documents the performance of three classifiers on two datasets, after having undergone the effects of three optimisation techniques. Section 4.1 explores the effects of data balancing and concludes if data balancing is necessary for the given datasets or not. Section 4.2 verifies the results provided by RF on PhiUSIIL and investigates why metrics were so high. Section 4.3 examines the effects feature selection and aims to cut down on features whilst maintaining or improving performance. Section 4.4 seeks to find optimal hyperparameters for each classifier to maximise F1 score. Section 4.5 performs final evaluations to assess the performance of optimised classifiers on the test set. Section 4.6 compares the proposed approach's performance to that of existing works.

4.1. Balanced Vs Unbalanced Data

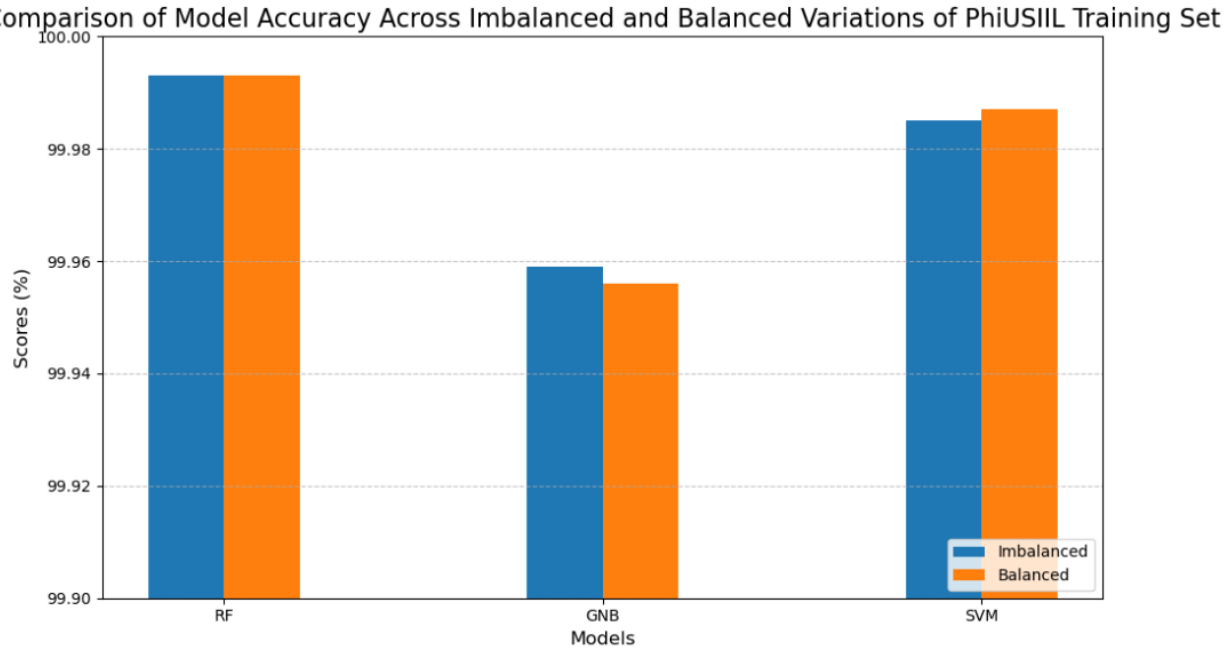
Imbalanced data can lead to bias of the majority class. Meaning, in binary classification, the machine learning model simply learns how to identify the majority label, rather than learning how to identify both majority and minority labels. Samad et al. (2023) found success in upsampling their datasets by creating synthetic samples using SMOTE. With a major catalyst for their decision to perform data balancing stemming from Zheng et al.'s (2022) work on how imbalanced data affects machine learning models. Which found that as imbalance rate increases, the classification accuracy decreases. Considering Samad et al.'s (2023) improvement in performance, data balancing was too explored in this work – also aiming to improve performance.

Dataset	**Before Upsampling**	**After Upsampling**
PhiUSIIL	Benign: 134,850 Phishing: 100,945	Benign: 134,850 Phishing: 100,945
ISCXURL-2016	Benign: 7,781 Phishing: 7,586	Benign: 7,781 Phishing: 7,781

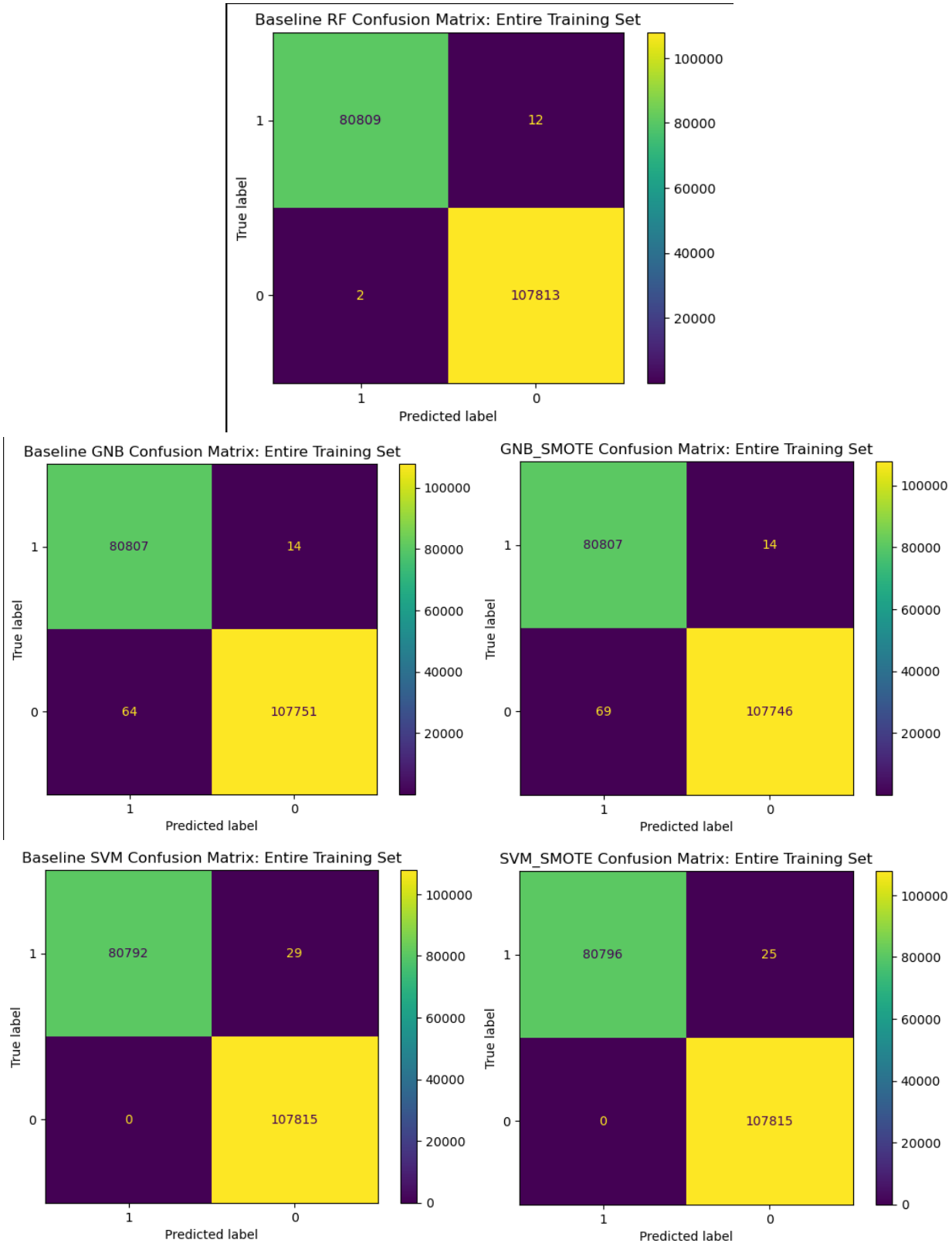
[Table 4 – Target class instances before and after upsampling]

Model	Type	Accuracy %	Precision %	Recall %	F1
RF	**Imbalanced**	**99.993**	**99.998**	**99.985**	**99.991**
	Balanced	99.993	99.998	99.985	99.991
GNB	**Imbalanced**	**99.959**	**99.921**	**99.983**	**99.952**
	Balanced	99.956	99.915	99.983	99.949
SVM	Imbalanced	99.985	100	99.964	99.982
	Balanced	**99.987**	**100**	**99.969**	**99.985**

[Table 5 – Classifier performances on balanced and imbalanced variations of PhiUSIIL]



[Figure 4 - Graphical comparison of classifier accuracy on balanced and imbalanced variations of PhiUSIIL]



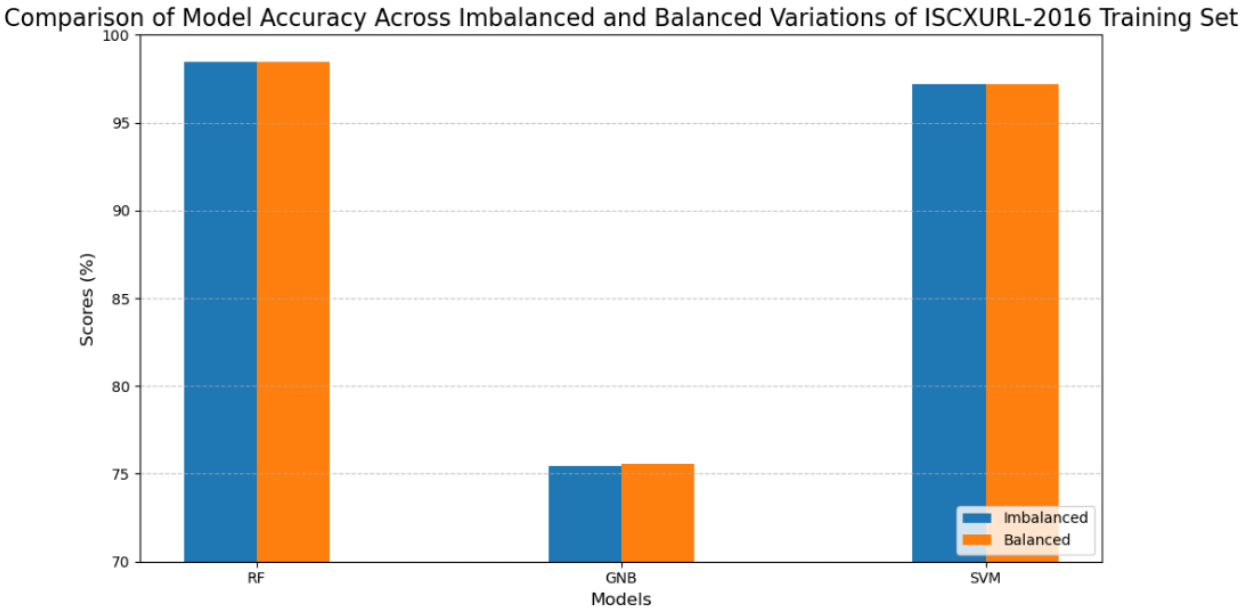
[Figure 5 – Comparison of Confusion Matrices of each classifier for unbalanced vs. balanced PhiUSIIL]

As made evident by the results in Table 5 and Figure 4, performance improvements and deterioration are negligible on PhiUSIIL. With data balancing applied, RF achieves the exact same performance, GNB's performance minutely worsens, and SVM slightly improves. Hence, the decision to not utilise data balancing for future analysis on PhiUSIIL.

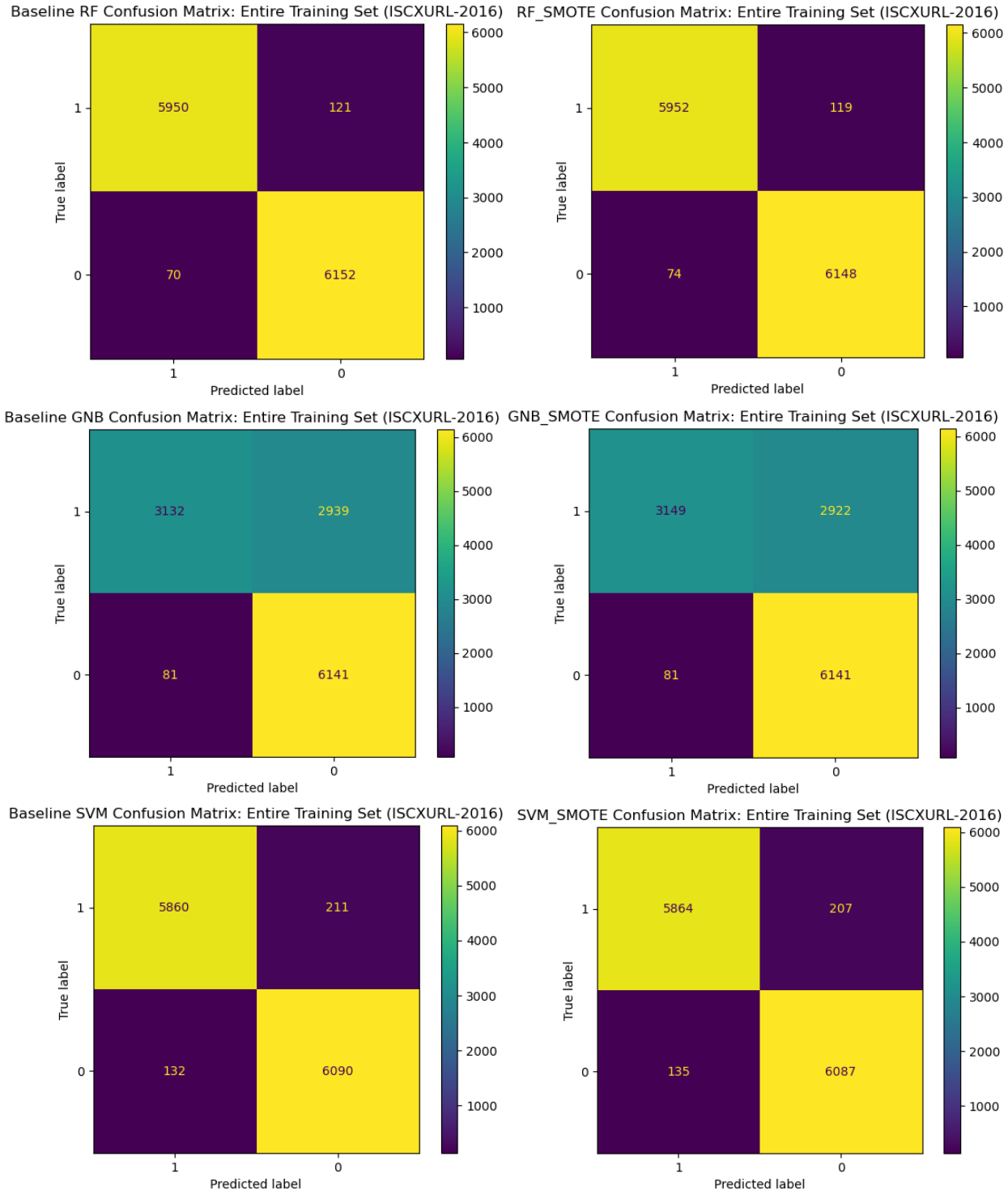
In reference to the baseline evaluations on imbalanced data in Table 5, and Figure 5, all classifiers exhibit excellent results with RF performing the best out of all the classifiers with a near-perfect accuracy of 99.993%. It is able to achieve both low FNs and FPs. Whereas, GNB relatively struggles with FPs, and SVM relatively struggles with FNs.

Model	Type	Accuracy %	Precision %	Recall %	F1
RF	**Imbalanced**	**98.446**	**98.839**	**98.007**	**98.421**
	Balanced	98.430	98.772	98.040	98.404
GNB	Imbalanced	75.433	97.459	51.590	67.386
	Balanced	**75.572**	**97.472**	**51.870**	**67.631**
SVM	Imbalanced	97.210	97.799	96.524	97.156
	Balanced	**97.218**	**97.751**	**96.590**	**97.166**

[Table 6 – Classifier performances on balanced and imbalanced variations of ISCXURL-2016]



[Figure 6 - Graphical comparison of classifier performances on balanced and imbalanced variations of ISCXURL-2016]



[Figure 7 – Comparison of Confusion Matrices of each classifier for unbalanced vs. balanced ISCXURL-2016]

Similar to the results on PhiUSIIL in Table 5 and Figures 5, the performance exhibited on the balanced version ISCXURL-2016 in Table 6 and Figure 7 indicate varying performance gains and losses of negligible magnitudes. After data balancing RF performs slightly worse, GNB performs slightly better, and SVM minutely improvess overall at the cost of trading precision for recall. Hence, the decision to not utilise data balancing for future analysis on ISCXURL-2016.

In reference to the baseline evaluations made on the imbalanced variant of ISCXURL-2016 in Table 6, and Figures 6 and 7, RF is also the best performing classifier with an accuracy of 98.446%. GNB exhibits extremely poor performance with an accuracy of 75.433, struggling heavily with FNs as made apparent in Figure 7. Indicating that it has learnt how to identify benign samples, but also has trouble learning how to identify phishing. Again, SVM relatively struggles with FNs, but also struggles with FPs. Exhibiting much higher FPs than RF in ISCXURL-2016 in Figure 7, compared to the FP comparison between SVM and RF in PhiUSIIL in Figure 6.

4.2. Verification of RF PhiUSIIL Performance

Following the near-perfect classification of the default RF model with an accuracy of 99.993% on the imbalanced variant of PhiUSIIL, further analysis had to be conducted to investigate how and why such excellent results were observed.

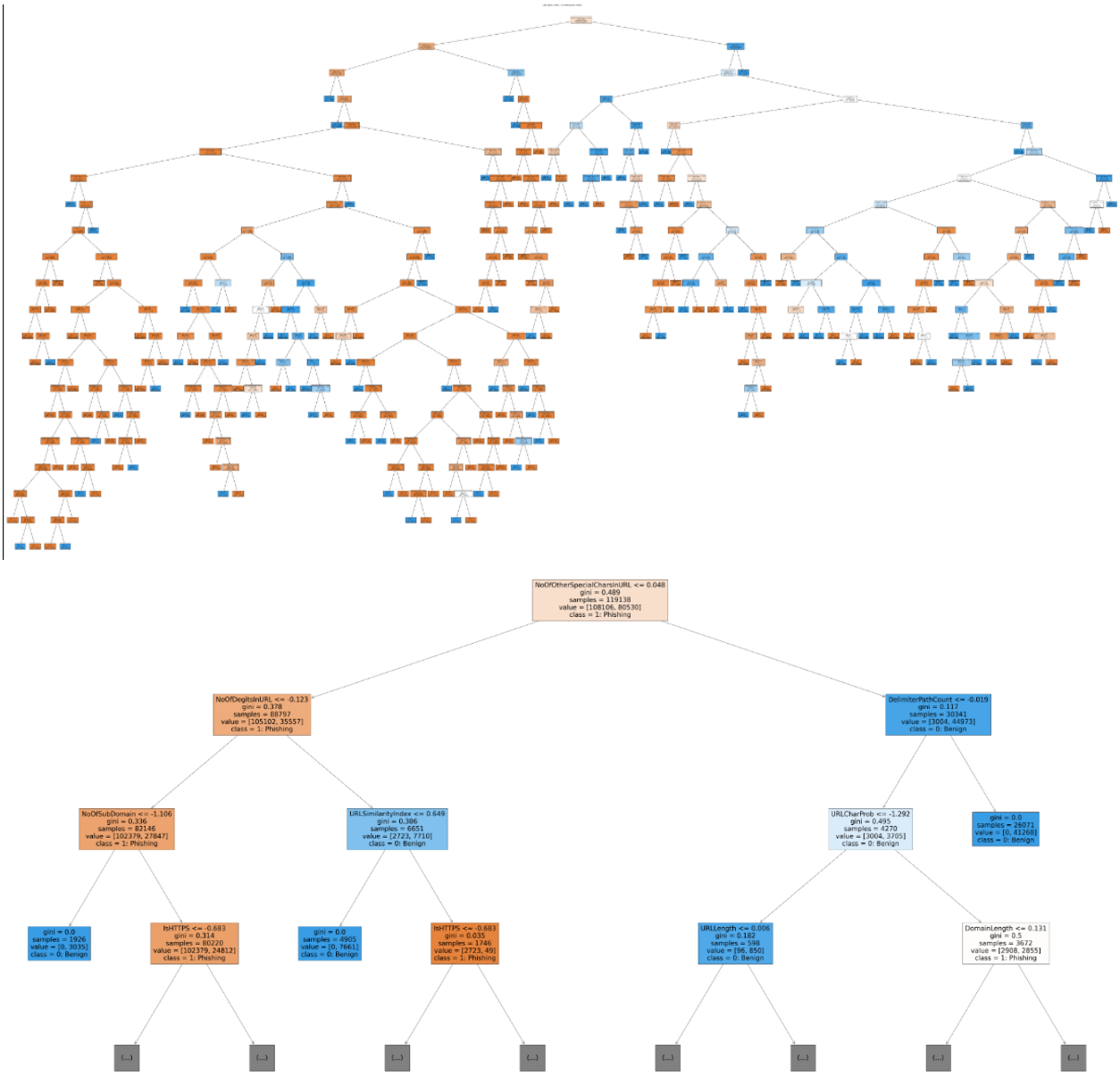
Firstly, evaluations on subsets of the training set of sizes 1000 and 100 were conducted using the RF model. The lower number of samples made it possible to increase the number of folds from 5 to 10 as execution of CV was quicker than if the whole training set was used.

Subset Size	Accuracy %	Precision %	Recall %	F1
1000	100	100	100	100
100	100	100	100	100

[Table 7 – RF performance on 1000 sample and 100 samples subsets of training set]

This validation check was utilised to see if there were any overfitting issues that would arise on the smaller subsets. However, perfect classification was still achieved.

Next, one of the DTs used in RF was plotted to observe and validate the decision making process.

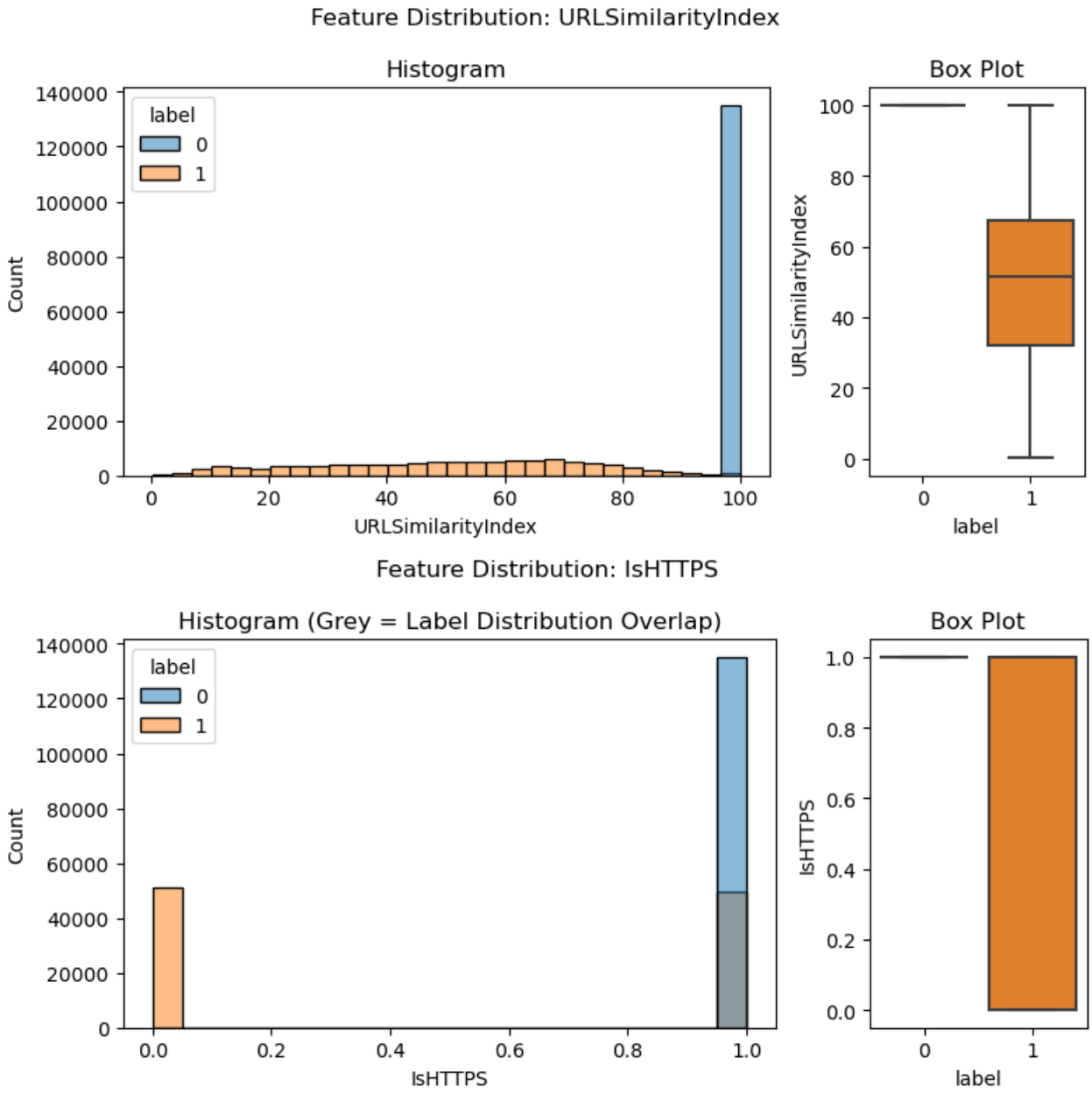


[Figure 8 – DT from default RF model. Whole tree vs. tree with maximum depth of 3]

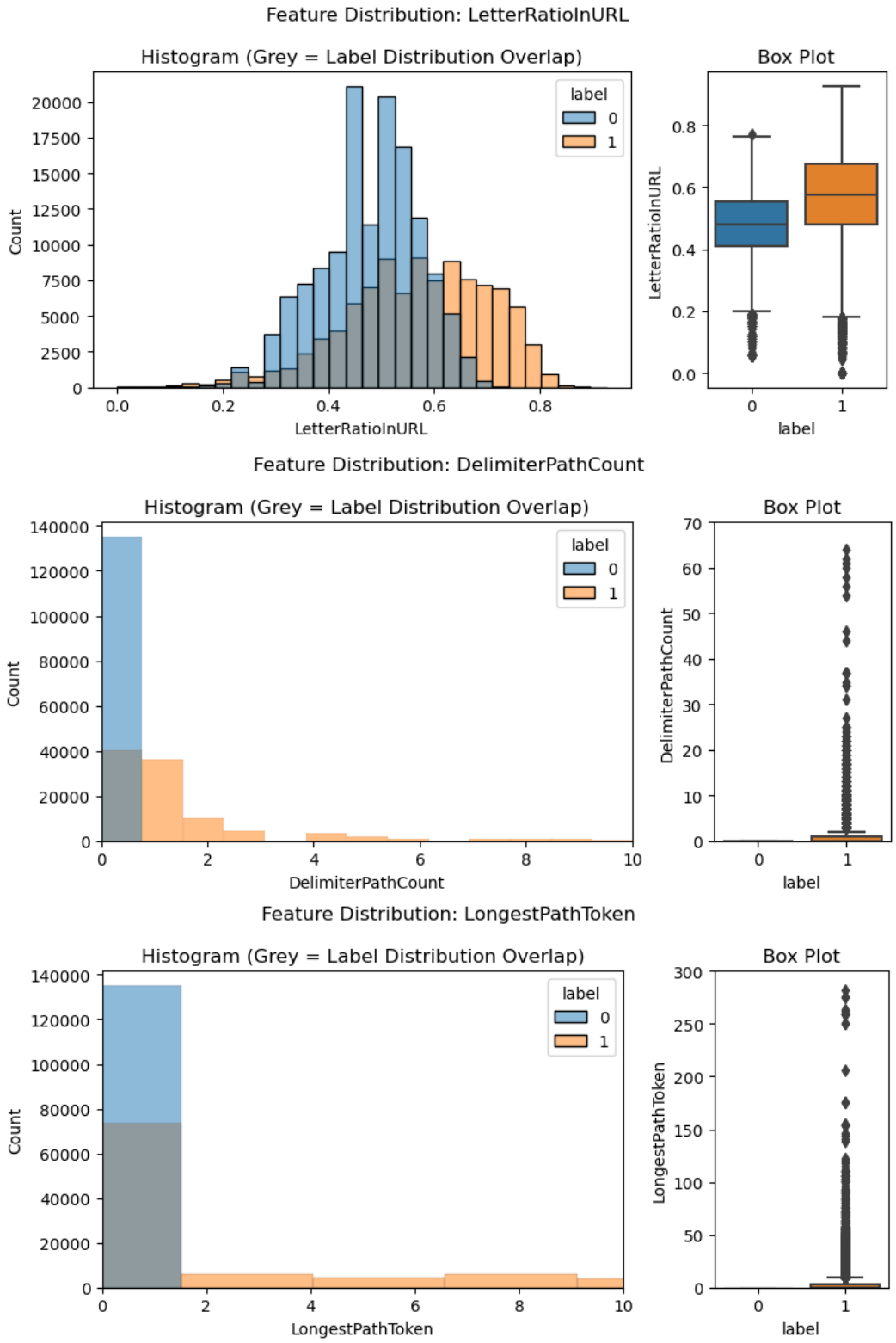
As seen in Figure 8, the depth of the DT is so large that it cannot be plotted in an interpretable manner. Thus, another zoomed in version was plotted with a maximum depth of 3 for interpretability. The more saturated the colour, the lower the gini. Meaning, lower impurity and mixing of classes at that node. Therefore, reflecting better decision-making and classification as samples in the node predominantly belong to a single class. This can be seen within the most saturated nodes on the zoomed in version of the tree, where on the left the 'IsHTTPS' node has a low gini of 0.035 with predominantly phishing samples.

Then on the right, the most pure nodes are the ‘DelimiterPathCount’ and ‘URLLength’ nodes. This then warranted more investigation into the features of PhiUSIIL.

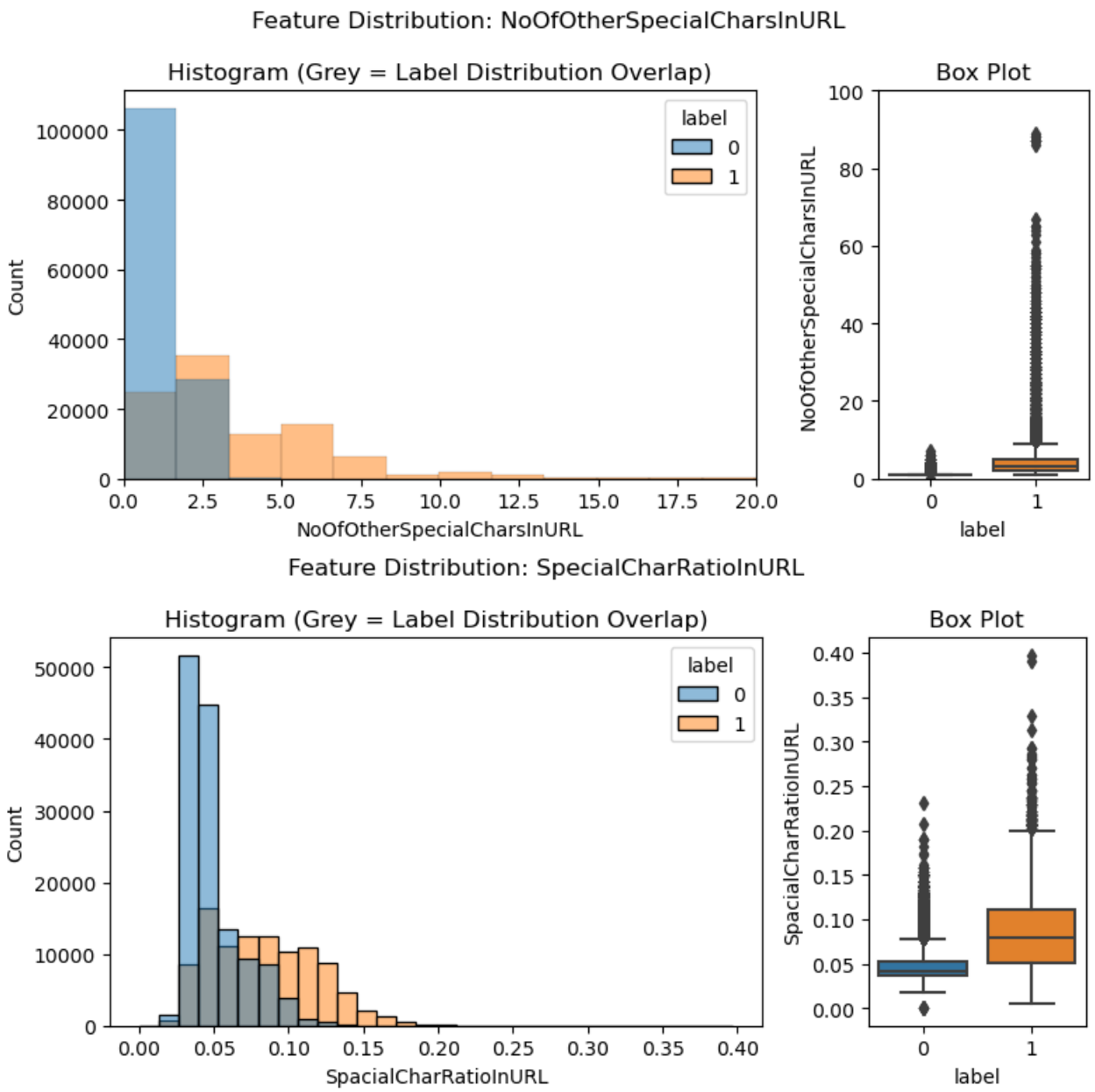
Hence, MI feature selection was conducted. To which then the feature distribution amongst target classes for features with an MI > 0.2 was examined. The features in question were: ‘URLSimilarityIndex’, ‘LetterRatioInURL’, ‘DelimiterPathCount’, ‘LongestPathToken’, ‘IsHTTPS’, ‘NoOfOtherSpecialCharsInURL’, and ‘SpecialCharRatioInURL’.



[Figure 9 - Feature Distribution of ’URLSimilairtyIndex’ and ’IsHTTPS’ features]



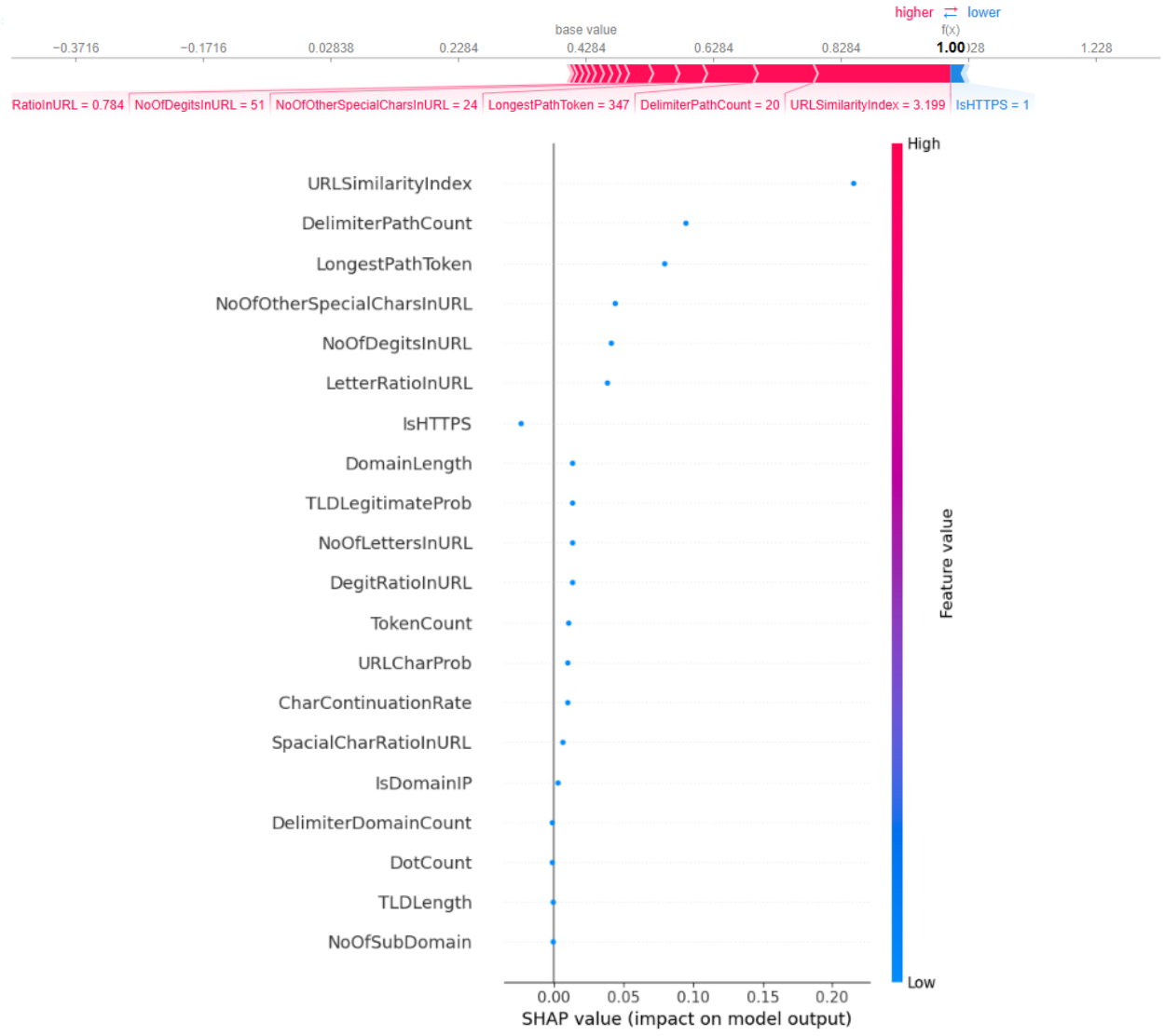
[Figure 10 - Feature Distribution of 'LetterRatioInURL', 'DelimiterPathCount', and 'LongestPathToken' features]



[Figure 11 - Feature Distribution of 'NoOfOtherSpecialCharsInURL' and 'SpecialCharRatioInURL' features]

The features in Figures 10 and 11 illustrate overlapping distribution of values amongst phishing and benign classes. However, Figure 9 exhibits two features, 'URLSimilarityIndex' and 'IsHTTPS' whose distribution clearly separates the phishing and benign classes. Therefore, giving a much clearer indication of why perfect classification was almost achieved with RF during the baseline evaluation.

To further verify these findings, the SHAP values for each feature in the PhiUSIIL dataset (the features remaining after preprocessing) were calculated on a single phishing sample. Essentially, this measures the importance of the feature in predicting whether an instance is phishing or benign. If the value is positive, the feature contributes to outputting the positive class (which in this case is 1 or phishing) and is coloured red. On the other hand, if the value is negative, then the feature contributes to the negative class (0 or benign) and is coloured blue. The further away from 0 the value is the, more impact that feature has in the classification of the sample.



[Figure 12 – Force Plot (above) and Summary Plot (below) of SHAP values for PhiUSIIL features]

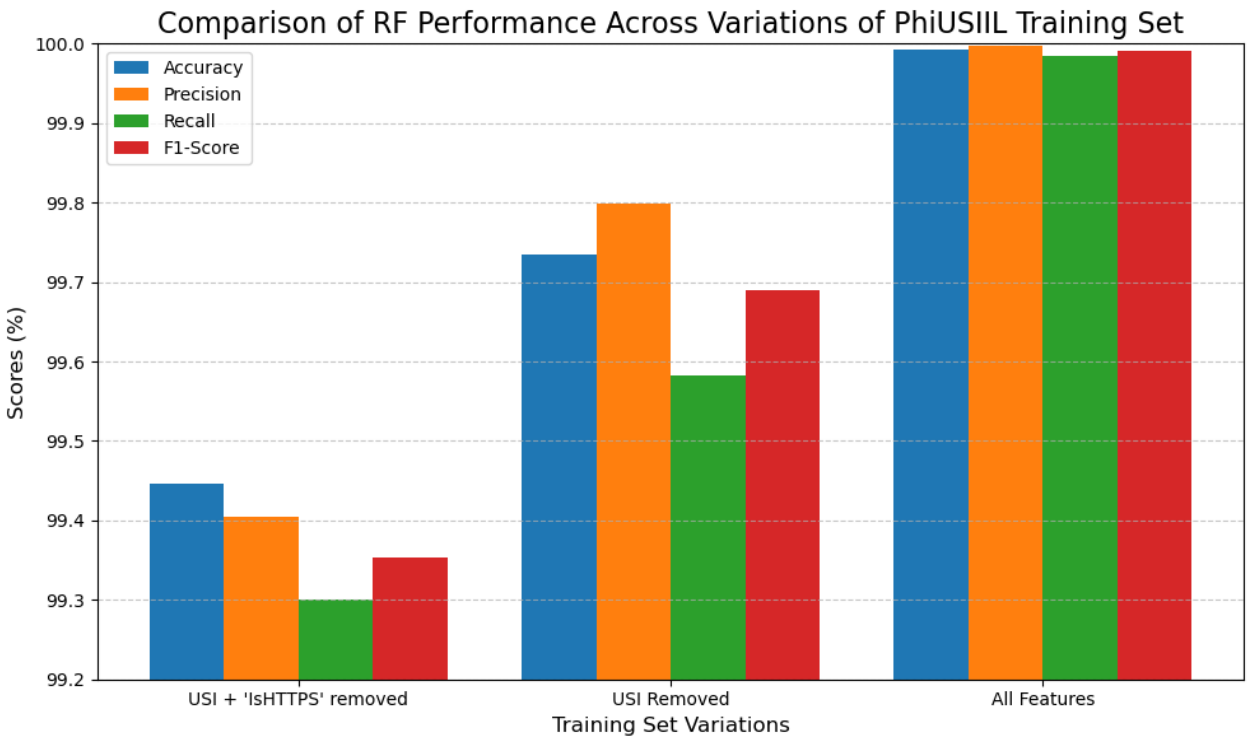
As seen in Figure 12, ‘URLSimilarityIndex’ has a SHAP value that is furthest from 0 out of all the features. Meaning, it had the most impact in classifying the sample as phishing. This makes sense as it had a low ‘URLSimilarityIndex’ value of 3.199, which is outside the range of values that benign samples have as can be seen in Figure 11. This coincides with Fajar et al.’s (2024) findings where via calculating SHAP values, they too concluded ‘URLSimilarityIndex’ as the most important feature for predicting the label.

Furthermore ‘IsHTTPS’ can be observed in attempting to classify the samples as benign due to the presence of the HTTPS protocol in the URL. However, its SHAP value is very close to 0, meaning it had little impact on classification. This corresponds to the distribution of ‘IsHTTPS’ in Figure 11 as there is still some overlap on the value 1 between phishing and benign samples. Despite a majority of samples being well separated.

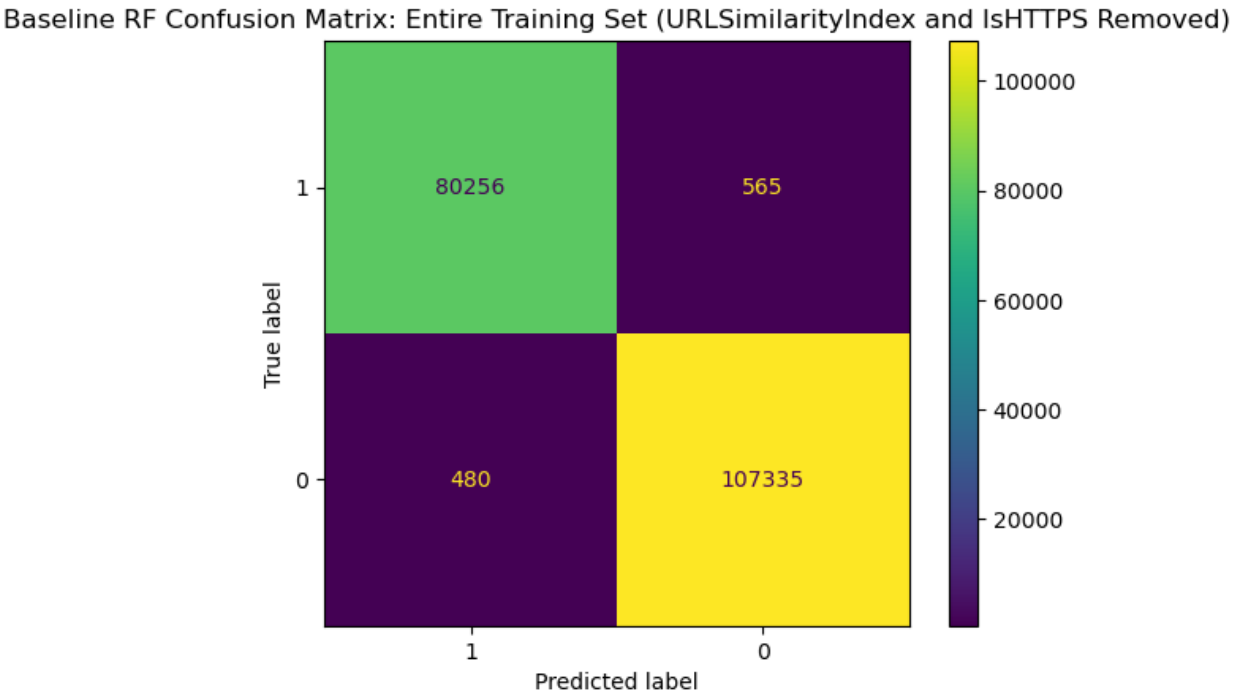
At this point, the most likely reason for RF’s near perfect classification stems from the clear distinction between class targets in the feature distribution of ‘URLSimilarityIndex’ (USI). To test this, the default RF model was evaluated on the training set again. However, this time USI was removed, and another test was conducted where both USI and ‘IsHTTPS’ was removed. Both were conducted using 5-fold CV.

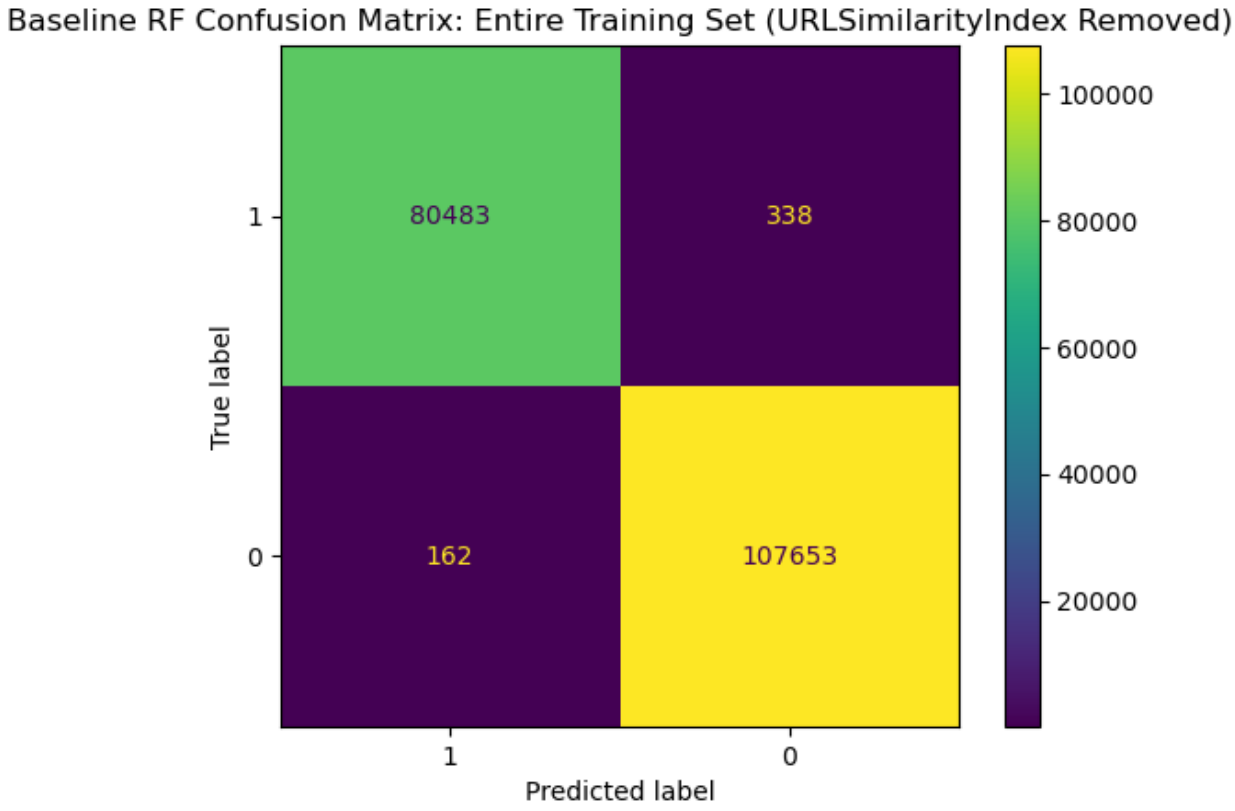
Features	Accuracy %	Precision %	Recall %	F1
USI + ‘IsHTTPS’ removed	99.446	99.405	99.301	99.353
USI removed	99.735	99.799	99.582	99.690
All	99.993	99.998	99.985	99.991

[Table 8 – Comparison of RF performance on PhiUSIIL training set with all features, USI removed, and USI and ‘IsHTTPS’ removed]



[Figure 13 - Graphical comparison of RF performance across variations of PhiUSIIL Training Set]





[Figure 14 - Comparison of Confusion Matrices of RF performances on PhiUSIIL when USI is removed, and when USI and 'IsHTTPS' are removed.]

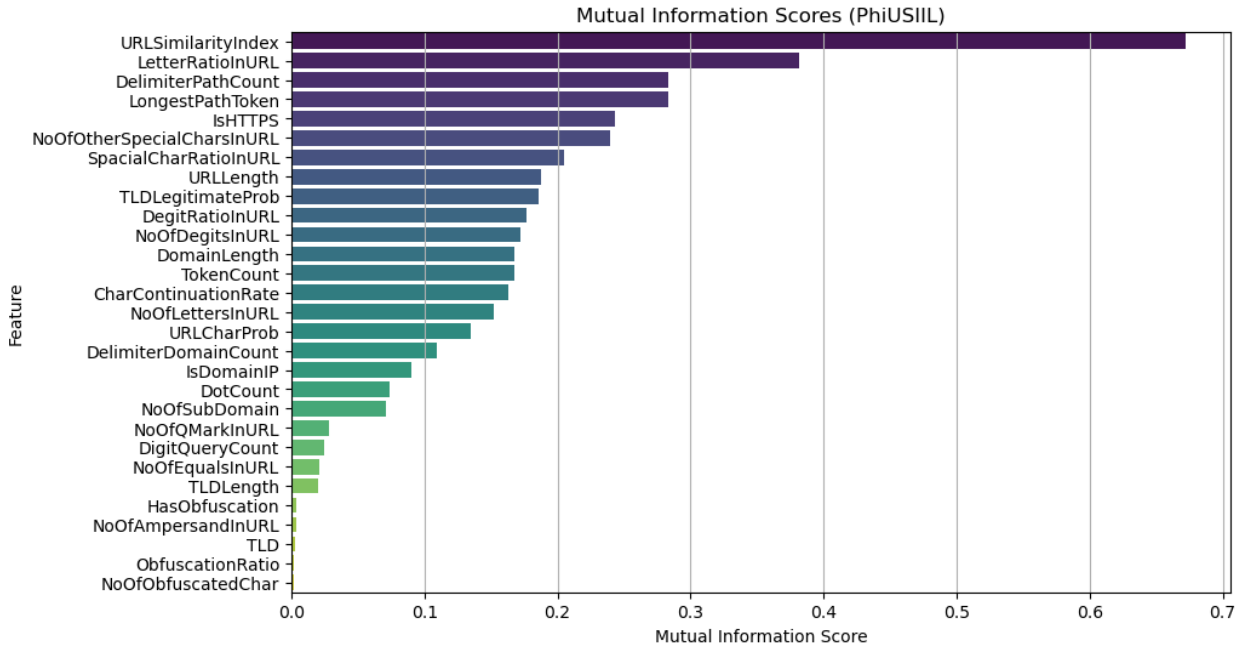
As expected in Table 8, and Figures 13 and 14, the performance of RF decreased. This lowered to an accuracy of 99.446% when USI was removed, and to 99.735% when both USI and 'IsHTTPS' was removed. These scores correspond to what is generally expected and seen within research where machine learning models are evaluated on datasets. With that being said, the performances are similar. However, the variation of PhiUSIIL with only USI removed exhibits better performance than that involving the removal of both USI and 'IsHTTPS'. Due to how polarising USI is, and how much more impactful it is than 'IsHTTPs' during classification (refer to Figure 12). Further analysis and evaluations will also be conducted on a variation of PhiUSIIL which does not contain USI. This is mainly to be used as a point of reference and comparison illustrating just how powerful of a feature USI is.

4.3. Feature Selection

Feature selection involves choosing a subset of features from the original dataset's features. It is commonly used to achieve benefits such as reducing redundancy, simpler understanding, faster training, simpler deployment, and reduced computational requirements (Samad et al., 2023; Rugangazi et al., 2023). A large number of variables can also create noise and reduce the accuracy of the model (Samad et al., 2023).

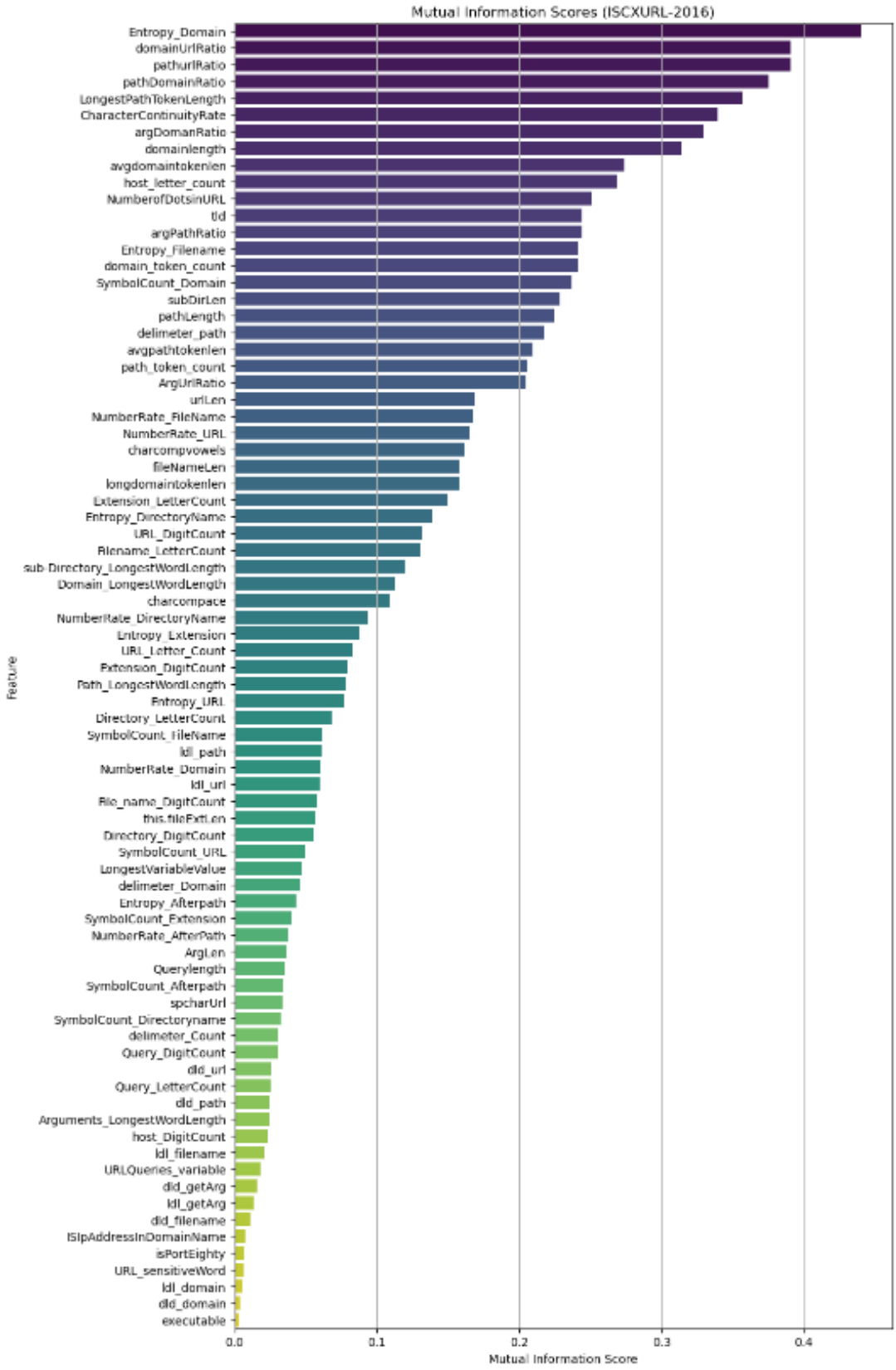
4.3.1. Mutual Information (MI)

MI feature selection was performed first as a general point of comparison for what features were to be deemed as important. By knowing the values of features in the dataset, it identifies the most useful features. This is achieved by assessing the amount of information learned about the target classes from the values of a given feature, which is reflected by its MI score (Vajrobol et al., 2024). Which means MI simply evaluates features independently of each other, contrary to evaluating them in the context of a specific model. Consequently, despite the MI scores of features, they may exhibit different behaviour depending on model and their interaction with other features in the dataset.



[Figure 15 - MI Scores for PhiUSIIL]

It can again be seen in Figure 15 that USI is a major contributor to the classification of samples in this dataset with its high MI score of 0.67. Meaning, it is highly informative of the target variable and has a strong dependency with it. Which coincides with Figure 12, and how indicative of phishing the USI of 3.199 is. Notice also that two of the extracted features ‘DelimiterPathCount’ and ‘LongestPathToken’ are amongst the top five most informative featurea. Whereas other features that were used by Gupta et al. (2021) like ‘TLDLength’ and ‘DigitQueryCount’ are considered to be some of the least informative.

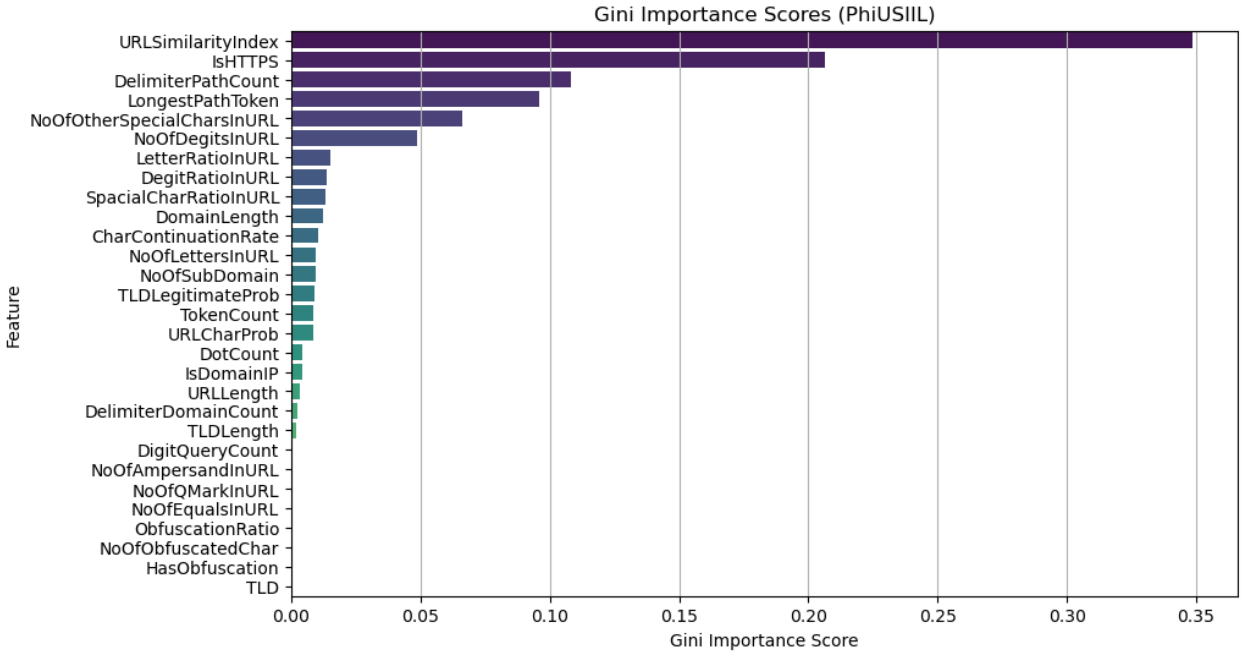


[Figure 16 – MI Scores for ISCXURL-2016]

Unlike in Figure 15, Figure 16 has no majorly informative feature that comes close to the calibre of USI. However, it can be observed that the features in Figure 16 are much more informative of their target variable than those in Figure 15, with there being 22 features with an MI > 0.2, compared to only 7 features having an MI > 0.2 in Figure 15.

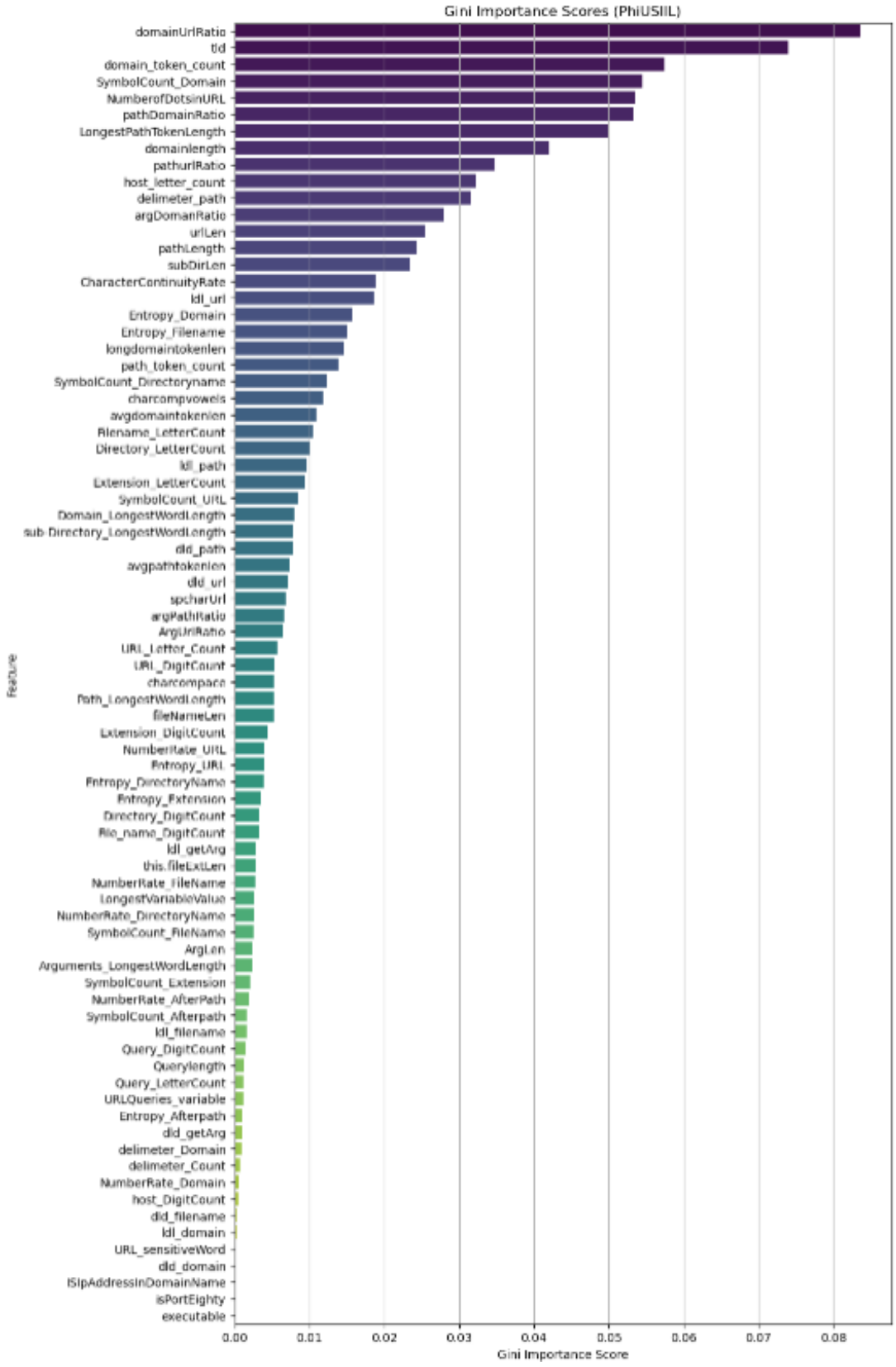
4.3.2. Random Forest

Mean Decrease in Impurity (MDI) feature selection was utilised to be a more tailored approach of feature selection for RF over MI. Mainly due to how it captures the behaviour of features within the context of RF, unlike MI. MDI evaluates the extent to which a feature contributes to reducing Gini impurity (GI) (Hong et al., 2016).



[Figure 17 - Gini Importance Scores for PhiUSIIL]

Despite being more informative than ‘IsHTTPS’ in Figure 15, ‘DelimiterPathCount’, ‘LongestPathToken’, and ‘LetterRatioInURL’ contribute significantly less than ‘IsHTTPS’ in reducing GI in Figure 17. Additionally, ‘LetterRatioInURL’ contributes significantly less than ‘DelimiterPathCount’ and ‘LongestPathToken’. ‘TLDLength’ and ‘DigitQueryCount’ are also considered to be amongst the least contributing features. Other extracted features that have low contributions also include ‘DelimiterDomainCount’, ‘URLLength’, and ‘DotCount’.



[Figure 17 - Gini Importance Scores for ISCXURL-2016]

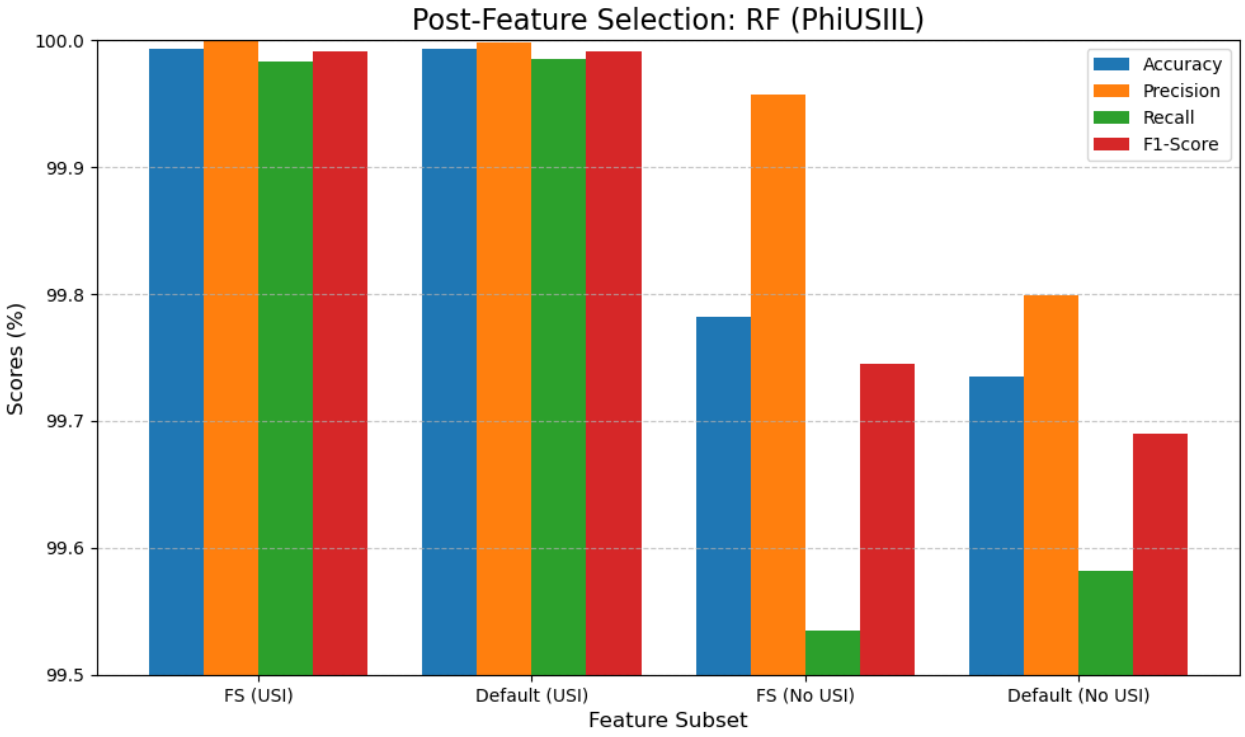
Despite both being modestly informative features in Figure 16 with an MI scores around 0.24, ‘tld’, ‘domain_token_count’, ‘SymbolCount_Domain’, and ‘NumberofDotsInURL’ are amongst the top 5 most contributive features in Figure 17.

Testing for the minimum number of features needed to achieve the same, or better performance than that of using all 30 features was centered around finding a GI threshold using Figure 17. The aim was to utilise as little of the most important features as possible to exhibit aforementioned performance.

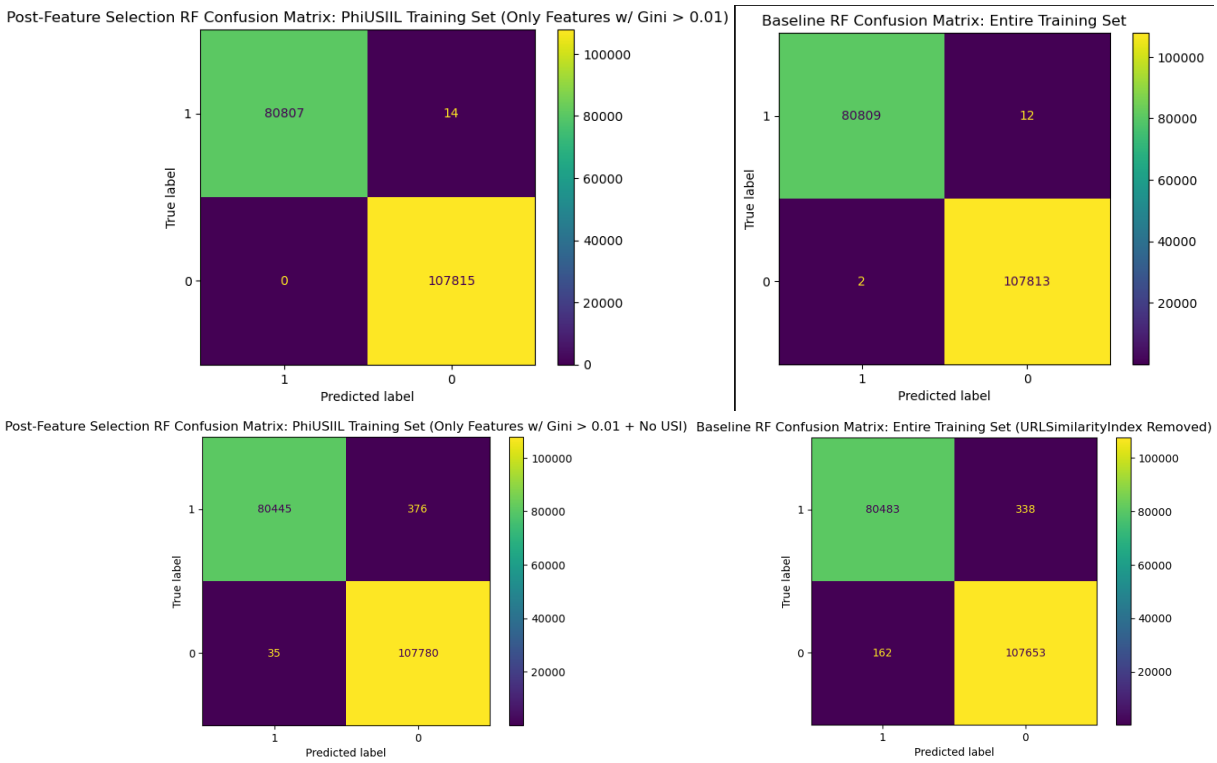
- FS: Feature selected subset of features.
- Default: All features of the dataset.

Feature	Type	No. of Features	Accuracy %	Precision %	Recall %	F1
USI	**FS**	**11**	**99.993**	**100**	**99.983**	**99.991**
	Default	30	99.993	99.998	99.985	99.991
No USI	**FS**	**10**	**99.782**	**99.957**	**99.535**	**99.745**
	Default	30	99.735	99.799	99.582	99.690

[Table 9 – Post-Feature Selection Evaluation: RF (PhiUSIIL)]



[Figure 18 – Graphical Representation of Post-Feature Selection Evaluation: RF (PhiUSIIL)]

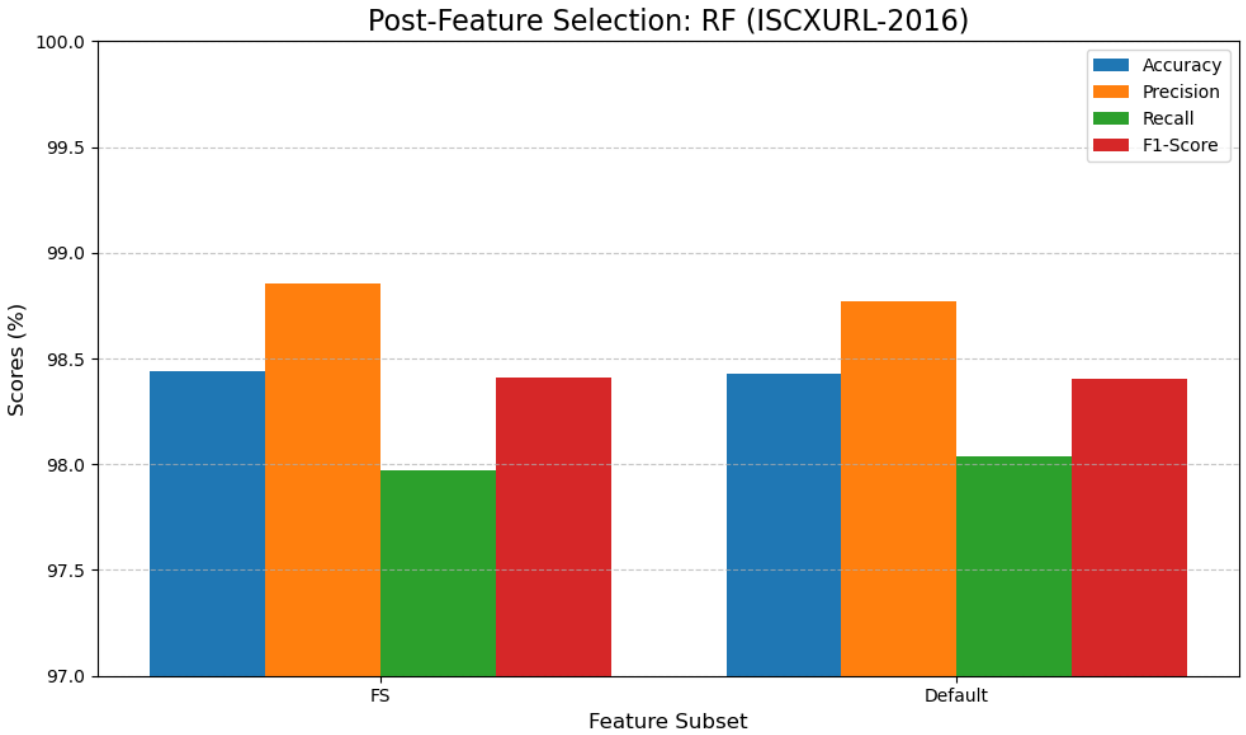


[Figure 19 – Post-Feature Selection Confusion Matrices: RF (PhiUSIIL)]

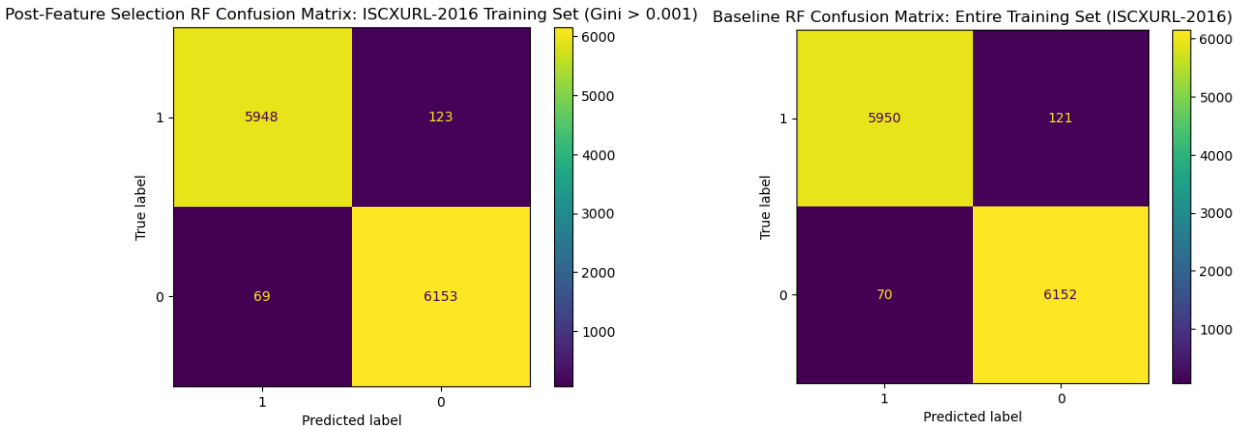
In Table 9, RF on PhiUSIIl (USI) achieved a performance similar to that of using all 30 features only using features with a Gini > 0.01 (11 features). While, PhiUSIIL (No USI) features with Gini > 0.01 (10 features) exhibited better accuracy of 99.782% and f1 of 99.745, at the cost of recall. Consequently, this feature subset was used for further analysis.

Type	No. of Features	Accuracy %	Precision %	Recall %	F1
FS	**67**	**98.438**	**98.854**	**97.974**	**98.412**
Default	78	98.430	98.772	98.040	98.404

[Table 10 – Post-Feature Selection Evaluation: RF (ISCXURL-2016)]



[Figure 20 – Graphical Representation of Post-Feature Selection Evaluation: RF (ISCXURL-2016)]



[Figure 21 – Post-Feature Selection Confusion Matrices: RF (ISCXURL-2016)]

In Table 10, using features with a Gini > 0.001 (67 features) exhibited similar performance to that of using all 79 features. With an increase in precision, in return for a minor decrease in recall. Consequently, this feature subset was used for further analysis.

4.3.3. Gaussian Naïve Bayes

SFS was performed to select optimal features for GNB. This was again mainly due to how features are evaluated in the context of a model, rather than individually like MI. SFS iteratively adds features and evaluates the model's performance afterwards. Then, selects the feature that provides the greatest improvement in the model's performance. By default, the target metric is accuracy. This occurs until no further improvement in performance is observed (Aggrawal & Pal, 2020).

Selected Features = 14	
URLLength	TLDLength
DomainLength	NoOfSubDomain
IsDomainIP	HasObfuscation
TLD	NoOfObfuscatedChar
URLSimilarityIndex	IsHTTPS
CharContinuationRate	DelimiterPathCount
TLDLegitimateProb	URLCharProb

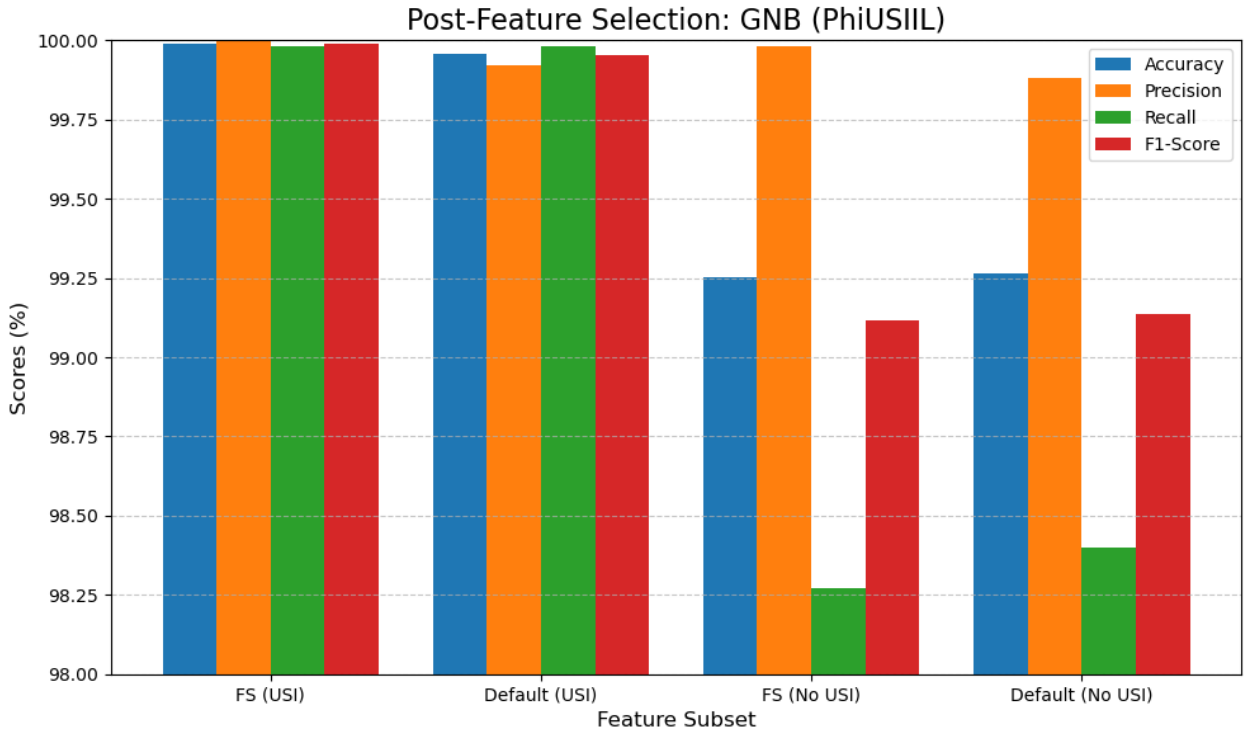
[Table 11 – Features selected by SFS: GNB (PhiUSIIL)]

Selected Features = 39	
domain_token_count	domainlength
charcompvowels	fileNameLen
charcompace	this.fileExtLen
ldl_url	pathurlRatio
ldl_domain	ArgUrlRatio
ldl_filename	argDomanRatio
urlLen	argPathRatio
NumberofDotsinURL	Directory_LetterCount
ISIpAddressInDomainName	Query_LetterCount
URL_DigitCount	LongestPathTokenLength
Directory_DigitCount	Domain_LongestWordLength
File_name_DigitCount	Path_LongestWordLength
URL_Letter_Count	spcharUrl
host_letter_count	delimeter_path
NumberRate_DirectoryName	Entropy_Domain
NumberRate_FileName	Entropy_DirectoryName
NumberRate_AfterPath	Entropy_Filename
SymbolCount_URL	Entropy_Extension
SymbolCount_Directoryname	Entropy_Afterpath
Entropy_URL	

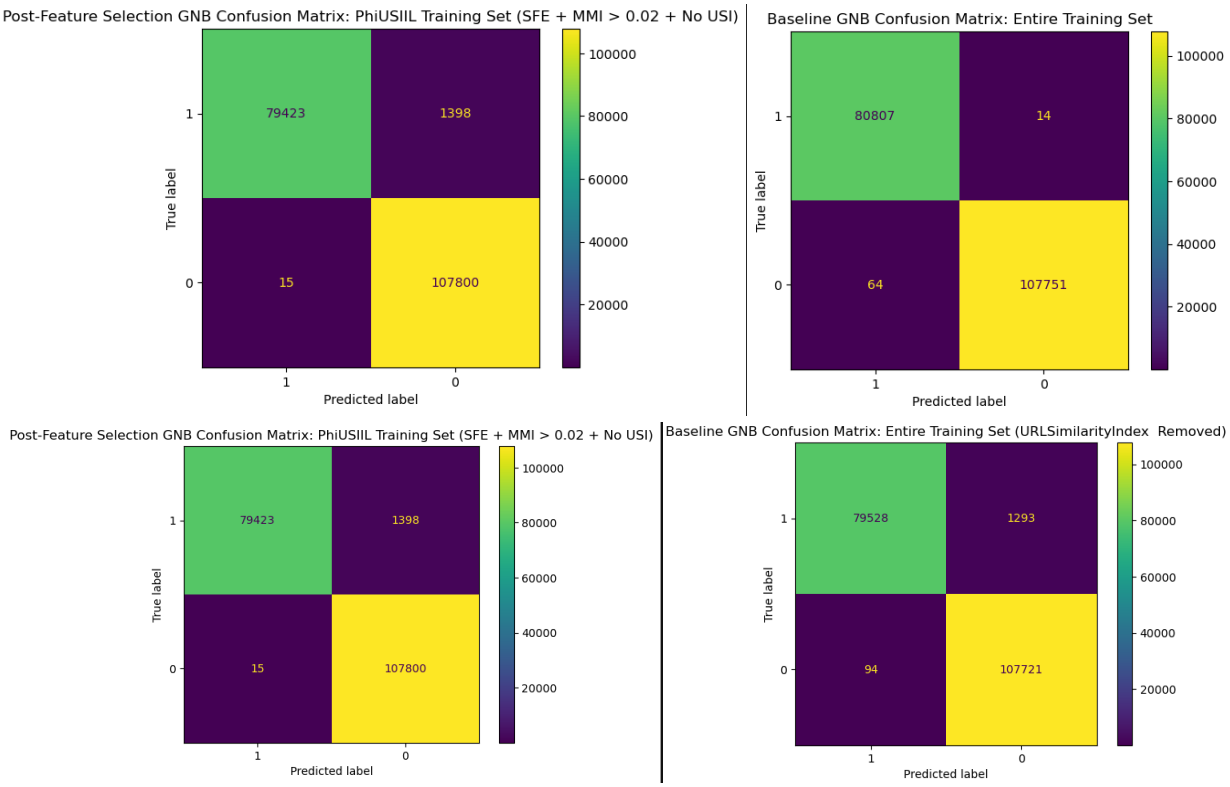
[Table 12 – Features selected by SFS: GNB (ISCXURL-2016)]

Feature	Type	No. of Features	Accuracy %	Precision %	Recall %	F1
USI	**FS**	**11**	**99.991**	**99.996**	**99.982**	**99.989**
	Default	30	99.959	99.921	99.983	99.952
No USI	**FS**	**10**	**99.251**	**99.981**	**98.270**	**99.118**
	Default	30	99.265	99.882	98.400	99.135

[Table 13 – Post-Feature Selection Evaluation: GNB (PhiUSIIL)]



[Figure 22 – Graphical Representation of Post-Feature Selection Evaluation: GNB (PhiUSIIL)]

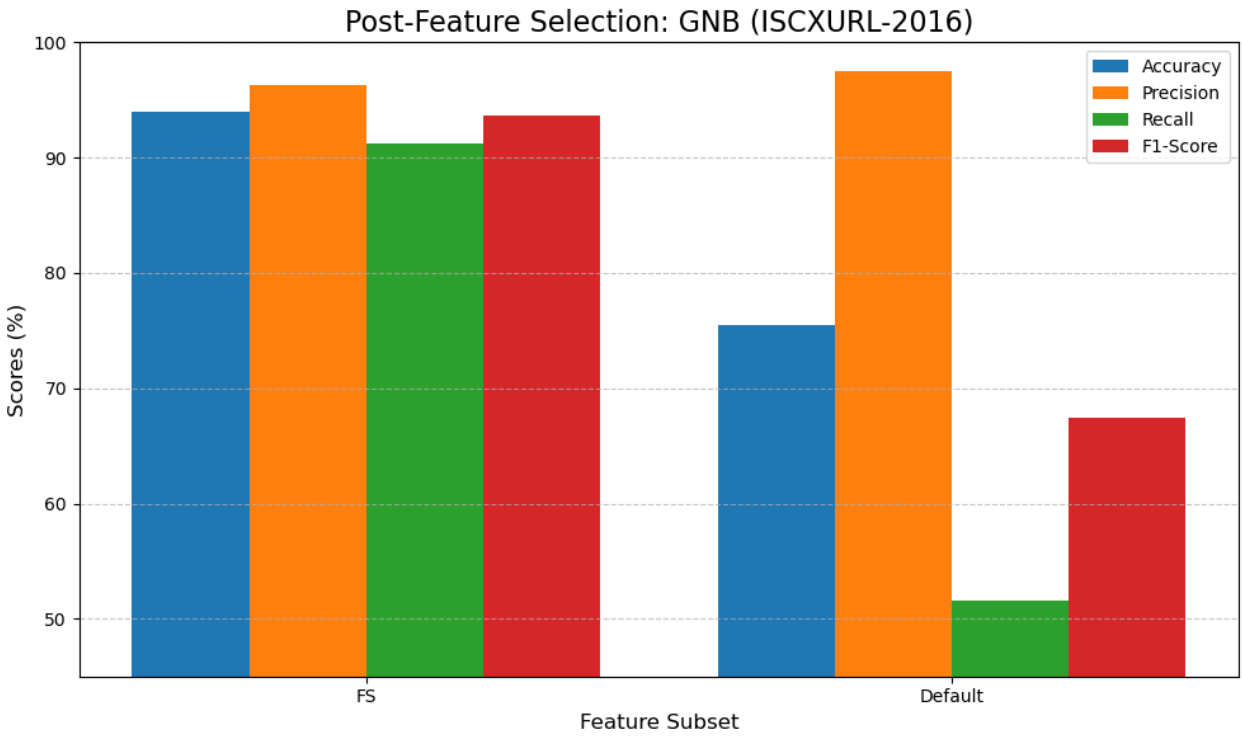


[Figure 23 - Post-Feature Selection Confusion Matrices: GNB (PhiUSIIL)]

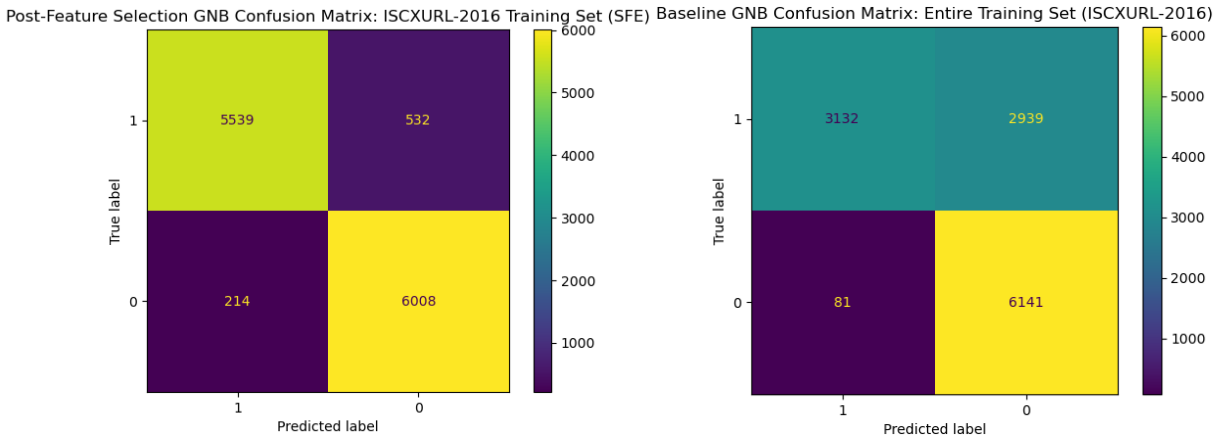
In Table 13, GNB on PhiUSIIl (USI), utilising the features selected by SFS that had an MI > 0.02 (11 features) achieved better performance than using all 30 features. For GNB on PhiUSIIL (No USI) SFS features with MI > 0.02 exhibited similar performance to if 30 features were used with a minor decrease in accuracy, recall and f1. However, this is a worthy sacrifice considering the sizeable improvement in precision jumping to 99.981%. Consequently, selected features were used for further analysis.

Type	No. of Features	Accuracy %	Precision %	Recall %	F1
FS	**39**	**93.932**	**96.279**	**91.237**	**93.690**
Default	78	75.433	97.459	51.590	67.386

[Table 14 – Post-Feature Selection Evaluation: GNB (ISCXURL-2016)]



[Figure 24 – Graphical Representation of Post-Feature Selection Evaluation: GNB (ISCXURL-2016)]



[Figure 25 - Post-Feature Selection Confusion Matrices comparison for GNB on ISCXURL-2016]

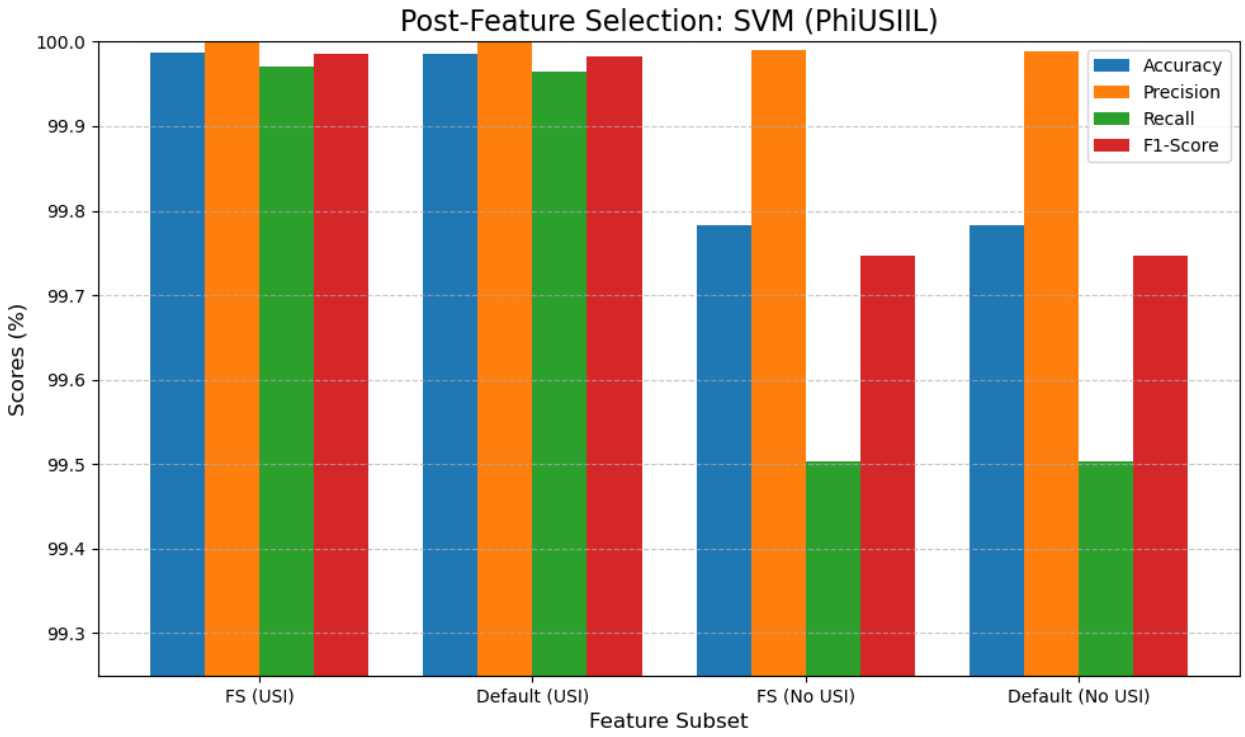
Table 14 and Figures 24 and 25 exhibit a tremendous improvement in performance after all 39 features selected by SFS were used, compared to when all 78 features are used. As a result, the SFS subset of features was utilised for further analysis.

4.3.4. Support Vector Machine

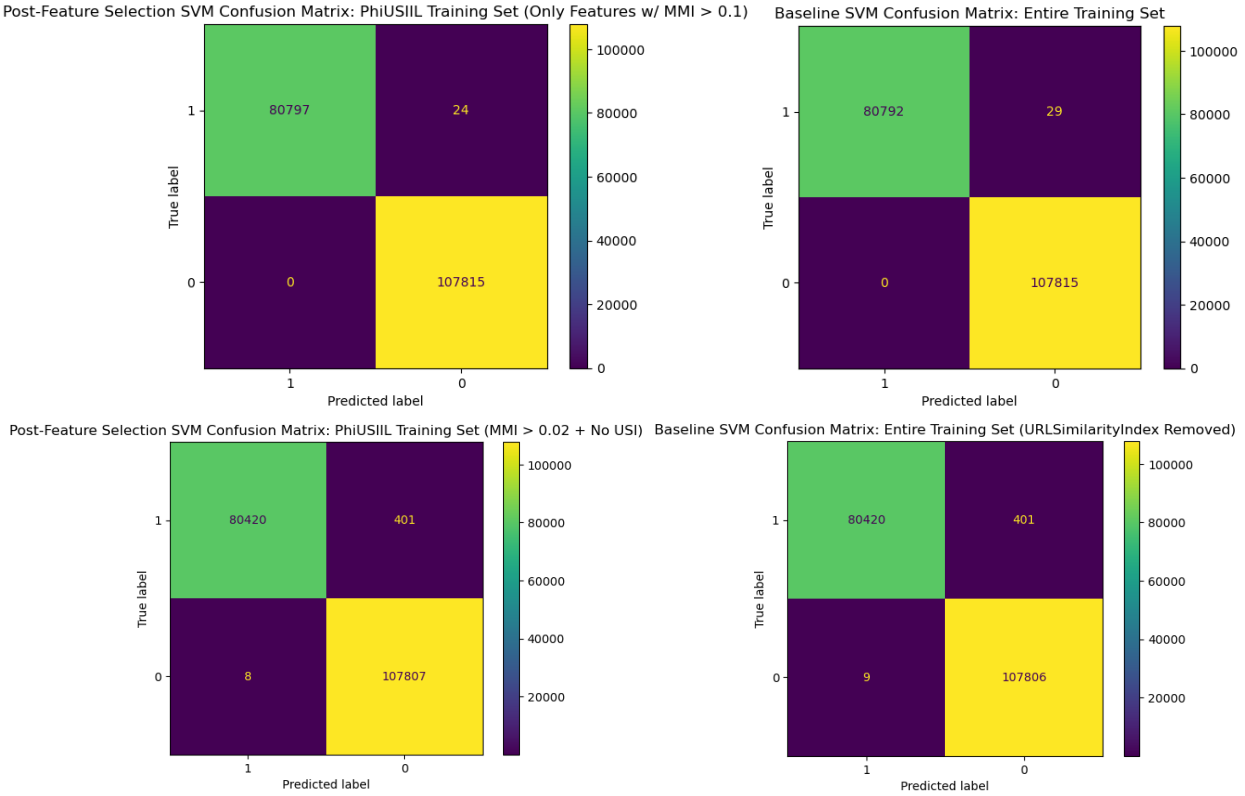
Originally, SFS was to be used as a tailored feature selection approach. However, the execution was unreasonably long due to hardware limitations. As a result, MI feature selection was utilised instead. The MI scores for PhiUSIIL and ISCXURL-2016 can be observed in Figures 15 and 16.

Feature	Type	No. of Features	Accuracy %	Precision %	Recall %	F1
USI	**FS**	**17**	**99.987**	**100**	**99.970**	**99.985**
	Default	30	99.985	100	99.964	99.982
No USI	**FS**	**13**	**99.783**	**99.990**	**99.504**	**99.746**
	Default	30	99.783	99.989	99.504	99.746

[Table 15 – Post-Feature Selection Evaluation: SVM (PhiUSIIL)]



[Figure 26 – Graphical Representation of Post-Feature Selection Evaluation: SVM (PhiUSIIL)]

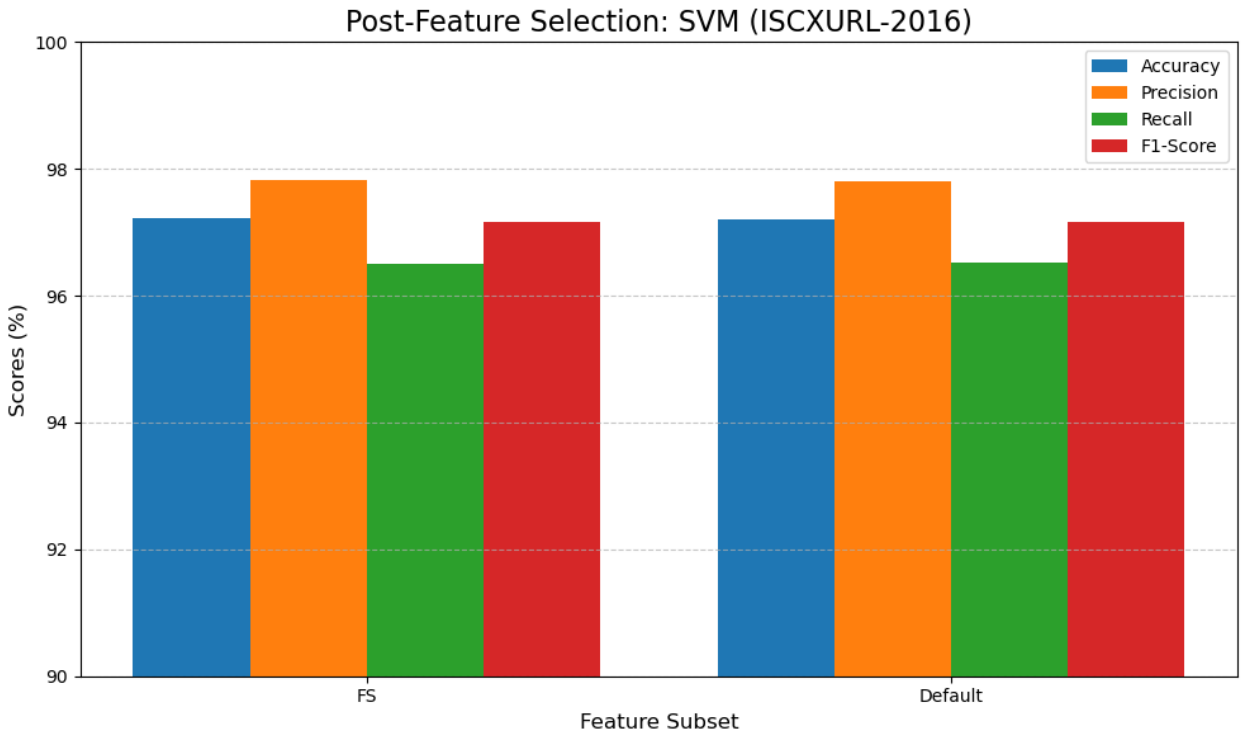


[Figure 27 - Post-Feature Selection Confusion Matrices: SVM (PhiUSIIL)]

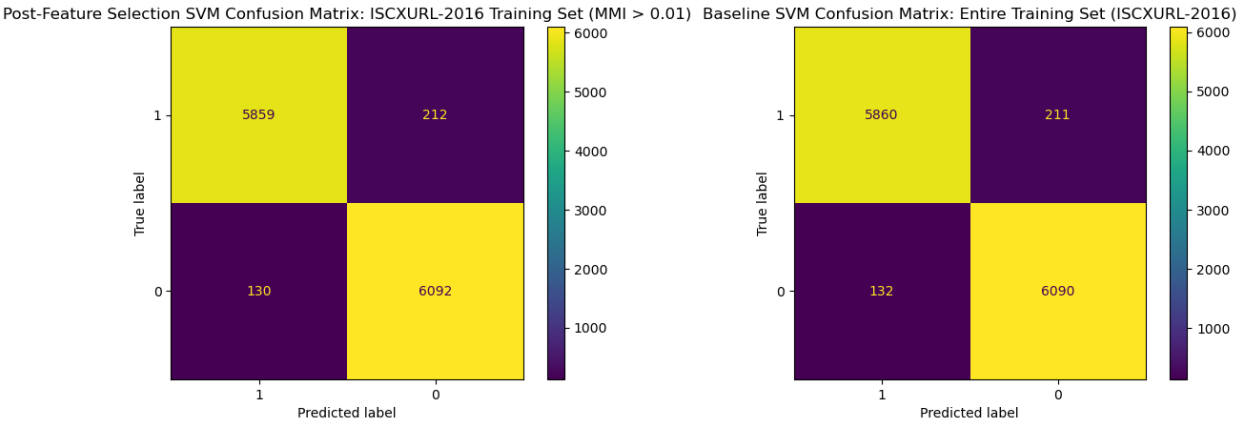
As seen in Table 15, and Figures 26 and 27, SVM evaluation on PhiUSIIL (USI) using features with MI > 0.1 (17 features) exhibited a minor increase in performance. Whereas, on PhiUSIIL (No USI), using features with MI > 0.02 results in almost the exact same performance as if all 30 features were used. Consequently, the selected feature subsets were used for further analysis.

Type	No. of Features	Accuracy %	Precision %	Recall %	F1
FS	**72**	**97.218**	**97.830**	**96.508**	**97.163**
Default	78	97.210	97.799	96.524	97.156

[Table 16 – Post-Feature Selection Evaluation: SVM (ISCXURL-2016)]



[Figure 28 – Graphical Representation of Post-Feature Selection Evaluation: SVM (ISCXURL-2016)]



[Figure 29 - Post-Feature Selection Confusion Matrices: SVM (ISCXURL-2016)]

Table 16 and Figure 23 illustrates similar performance to using all 78 features when features with an MI > 0.01 are utilised (72 features). As a result, the selected feature subset was used for further analysis.

4.4. Hyperparameter Tuning

Hyperparameter tuning is commonly utilised to improve the performance of a model. Samad et al., (2023) attribute the largest improvement in the performance of the models in their study to hyperparameter tuning. As a result, automatic hyperparameter tuning using GridSearchCV was used with the target metric to optimise being F1. Validation curves were also used to scout which values should be used within grid search. However, validation curves were mainly used to tune SVM as GridSearchCV and RandomisedSearchCV exhibited unreasonable durations of execution.

Classifier	Hyperparameters	PhiUSIIL (USI) Values	PhiUSIL (No USI) Values	ISCXURL-2016 Values
RF	max_features	Sqrt	Sqrt	Sqrt
	n_estimators	100	100	100
	max _depth	5	20	20
	min_samples_split	2	5	3
	min_samples_leaf	1	1	1
GNB	var_smoothing	1e-12	1e-4	1e-2
SVM	C	3	10	4
	gamma	Scale	Scale	0.1
	kernel	Linear	rbf	rbf

[Table 17 – Optimal hyperparameters for RF, GNB, and SVM for PhiUSIIL and ISCXURL-2016.]

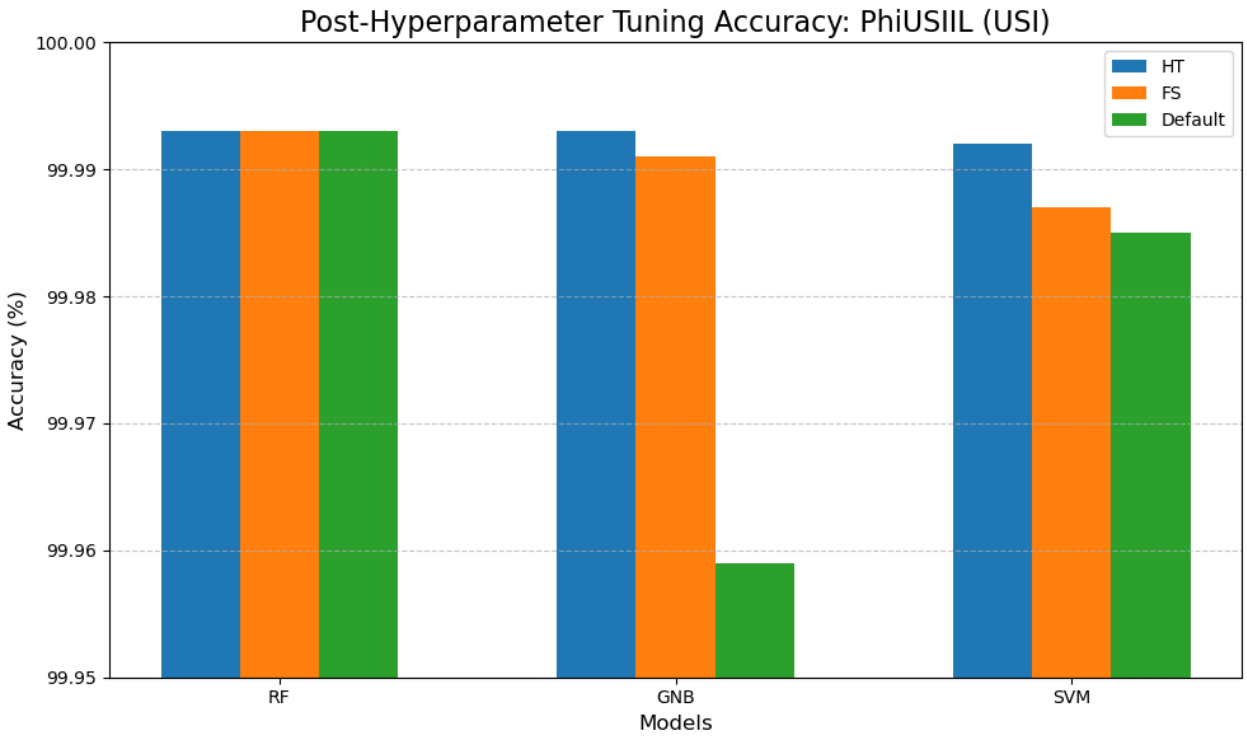
The hyperparameter tuned models were then evaluated on the training set utilising 5-fold CV.

- HT: Hyperparameter tuned models trained and tested on the feature selected subset of features corresponding to the respective model.
- FS: Default models trained and tested on the feature selected subset of features.
- Default: Default models trained and tested on all features of the training set.

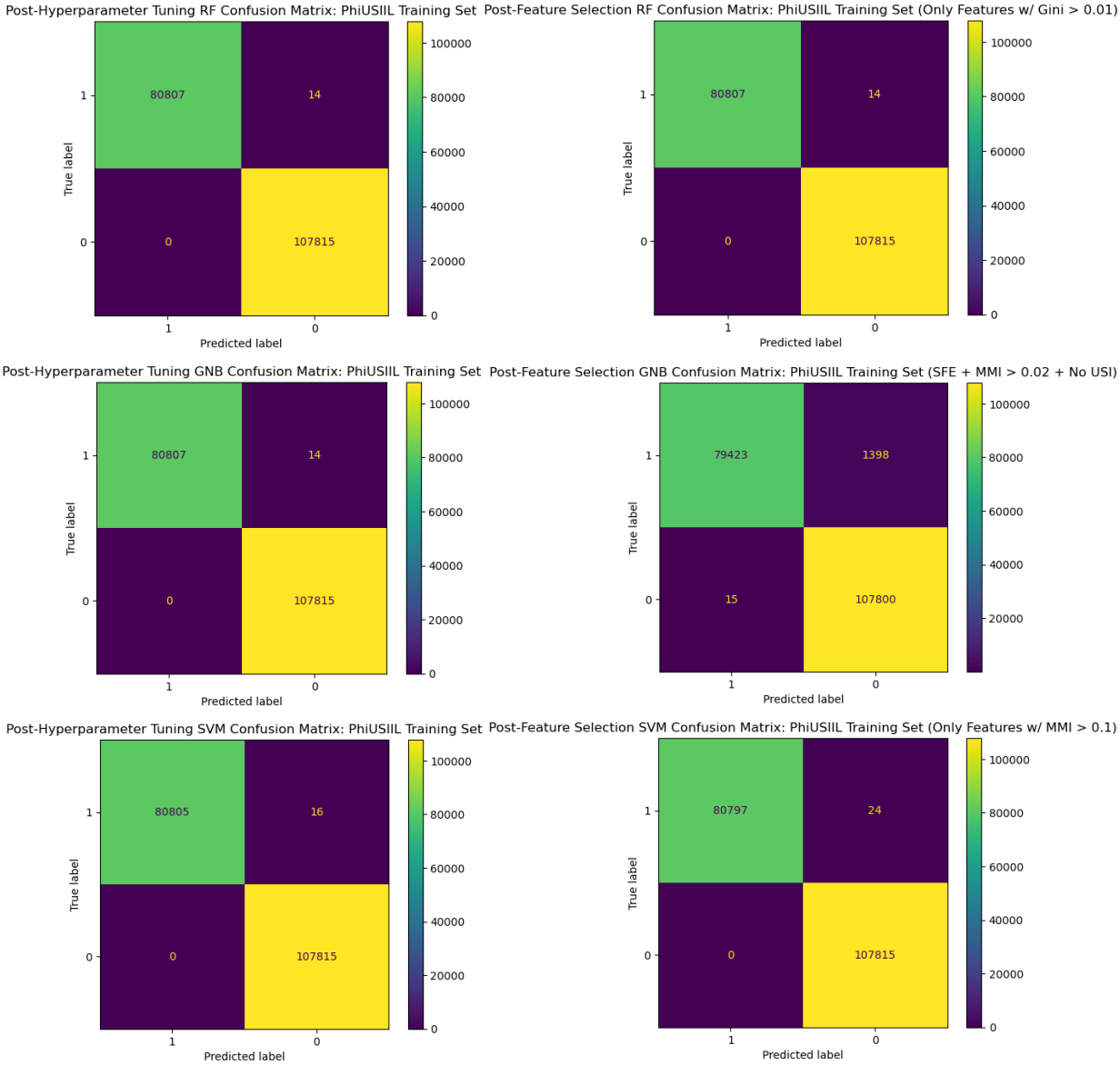
4.4.1. PhiUSIIL (USI)

Model	Type	Accuracy %	Precision %	Recall %	F1
RF	**HT**	**99.993**	**100**	**99.983**	**99.991**
	FS	99.993	100	99.983	99.991
	Default	99.993	99.998	99.985	99.991
GNB	**HT**	**99.993**	**100**	**99.983**	**99.991**
	FS	99.991	99.996	99.982	99.989
	Default	99.959	99.921	99.983	99.952
SVM	**HT**	**99.992**	**100**	**99.980**	**99.990**
	FS	99.987	100	99.970	99.985
	Default	99.985	100	99.964	99.982

[Table 18 – Post-Hyperparameter Tuning Evaluation and Comparison: PhiUSIIL (USI)]



[Figure 30 - Post-Hyperparameter Tuning Evaluation and Comparison: Accuracy PhiUSIIL (USI)]



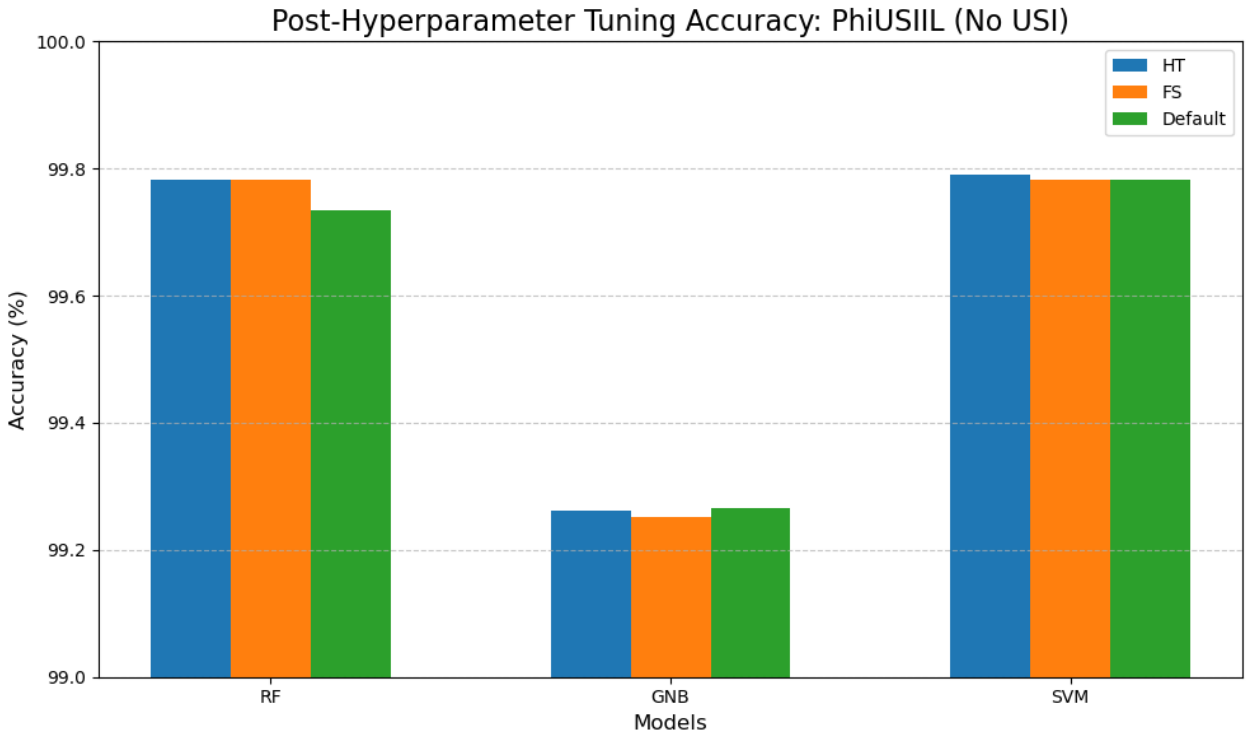
[Figure 31 - Post-Hyperparameter Tuning Confusion Matrices: PhiUSIIL (USI)]

Table 18 shows that hyperparameter tuning made no difference on the performance of RF. However, GNB and SVM exhibit a minor increase in performance. Ultimately resulting in each models performing very similar to each other, with RF and GNB sharing the exact same performance across all metrics. Thereby making them best overall classifiers for the PhiUSIIL (USI) training set with an accuracy of 99.993%.

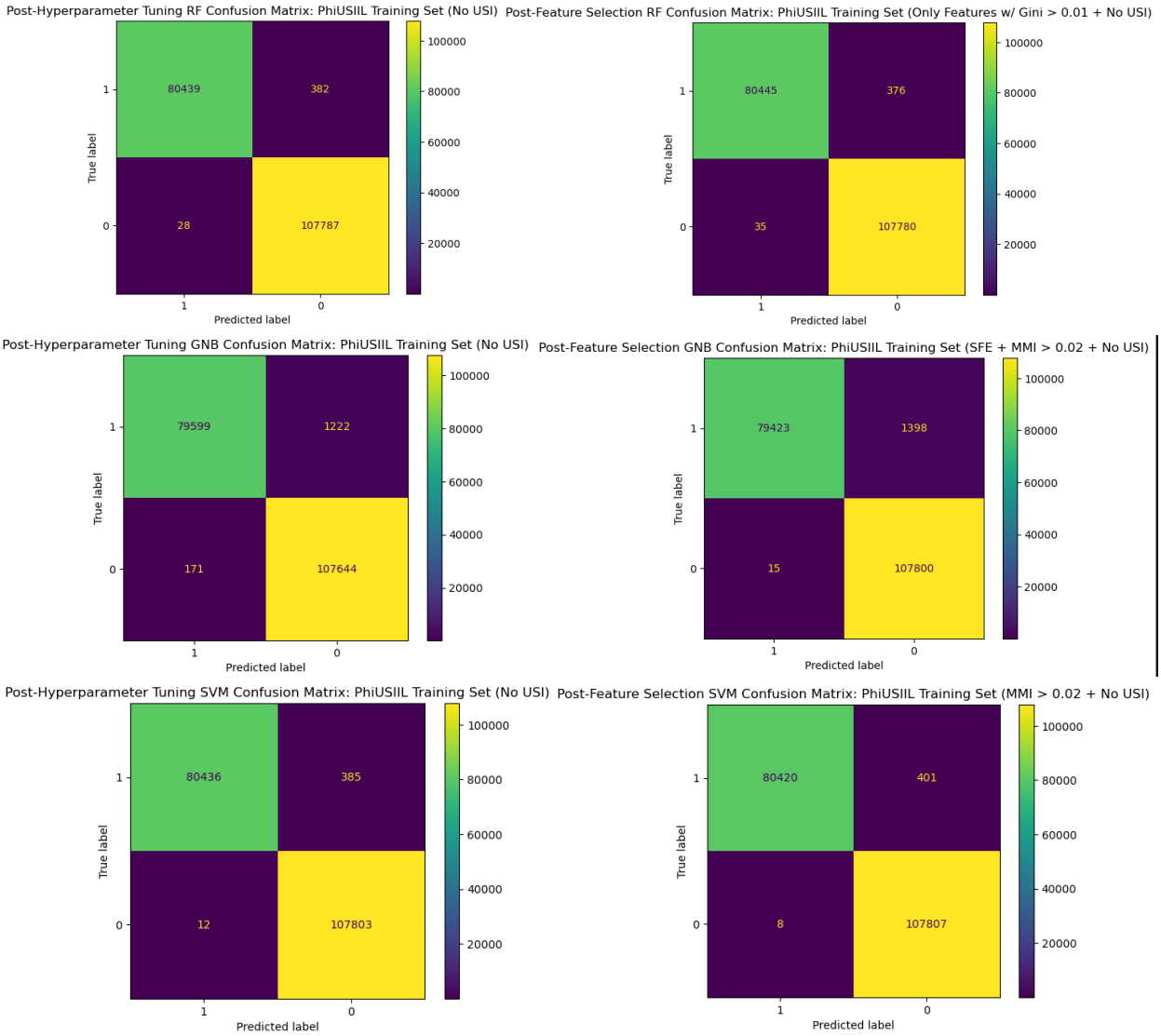
4.4.2. PhiUSIIL (No USI)

Model	Type	Accuracy %	Precision %	Recall %	F1
RF	**HT**	**99.783**	**99.965**	**99.527**	**99.745**
	FS	99.782	99.957	99.535	99.745
	Default	99.735	99.799	99.582	99.690
GNB	HT	99.262	99.786	98.488	99.133
	FS	99.251	99.981	98.270	99.118
	Default	**99.265**	**99.882**	**98.400**	**99.135**
SVM	**HT**	**99.790**	**99.985**	**99.524**	**99.754**
	FS	99.783	99.990	99.504	99.746
	Default	99.783	99.989	99.504	99.746

[Table 19 – Post-Hyperparameter Tuning Evaluation and Comparison: PhiUSIIL (No USI)]



[Figure 32 - Post-Hyperparameter Tuning Evaluation and Comparison: Accuracy PhiUSIIL (No USI)]



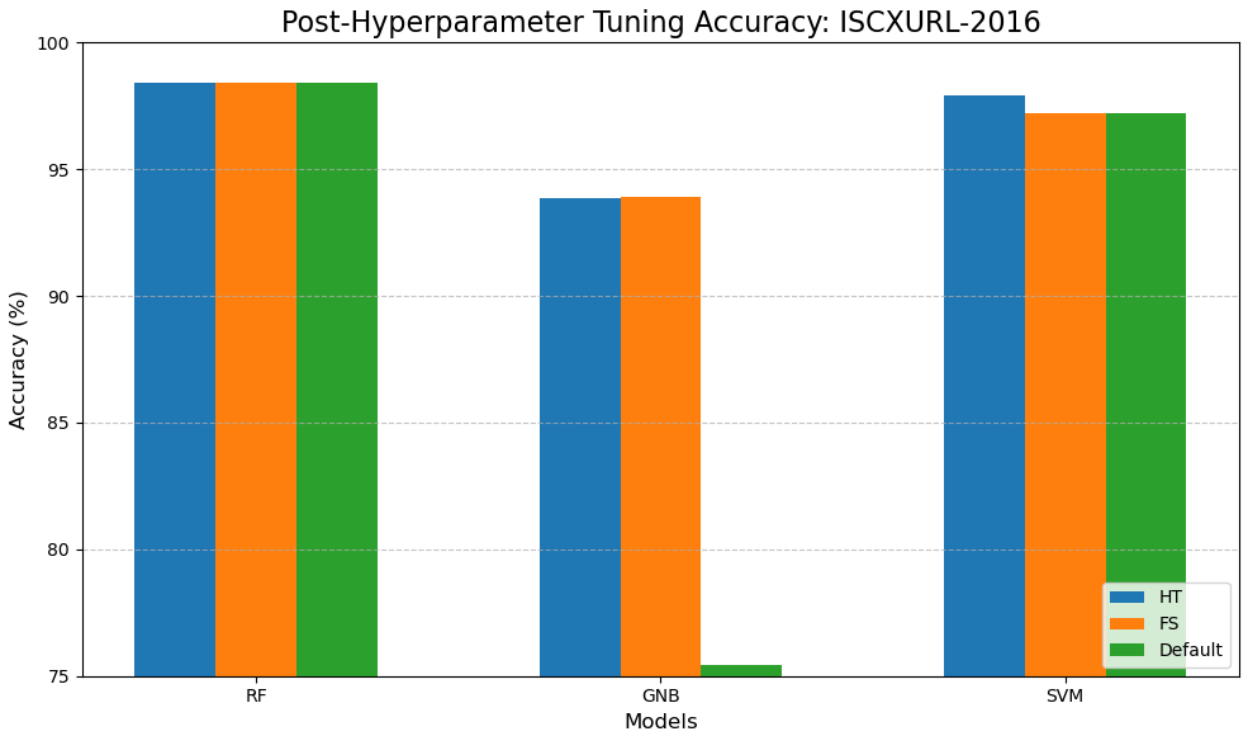
[Figure 33 - Post-Hyperparameter Tuning Confusion Matrices: PhiUSIIL (No USI)]

Table 19 sees SVM perform as the best overall classifier for the PhiUSIIL (No USI) training set with an accuracy of 99.790%. RF performs similarly after hyperparameter tuning, to how it performed without hyperparameter tuning on the feature selected subset. On the other hand, hyperparameter tuning has minorly increased the performance of GNB when compared to its exclusively feature selected counterpart, minus the drop in precision down to 99.786%. However, overall the tuned model performs slightly worse than the default model using all features in the dataset, with its lower scores except for recall at 98.488%, compared to FS recall which is 98.400. For SVM, hyperparameter tuning has modestly increased performance with an accuracy of 99.790%. The only drawback being the minute drop in precision down to 99.985% from 99.990%.

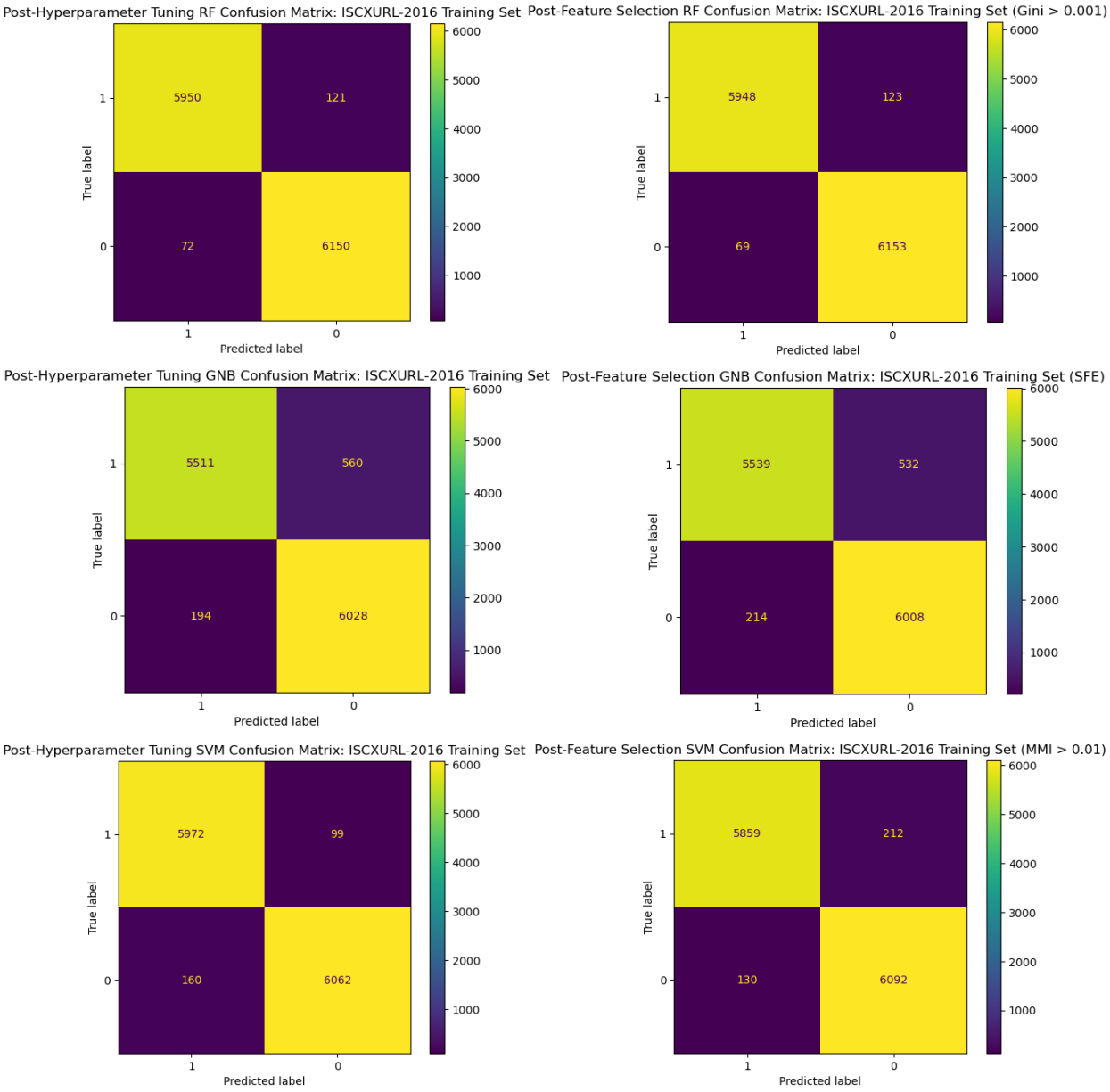
4.4.3. ISCXURL-2016

Model	Type	Accuracy %	Precision %	Recall %	F1
RF	HT	98.430	98.807	98.006	98.404
	FS	**98.438**	**98.854**	**97.974**	**98.412**
	Default	98.430	98.772	98.040	98.404
GNB	HT	93.866	96.599	90.776	93.596
	FS	**93.932**	**96.279**	**91.237**	**93.690**
	Default	75.433	97.459	51.590	67.386
SVM	**HT**	**97.893**	**97.393**	**98.369**	**97.878**
	FS	97.218	97.830	96.508	97.163
	Default	97.210	97.799	96.524	97.156

[Table 19 – Post-Hyperparameter Tuning Evaluation and Comparison: ISCXURL-2016]



[Figure 34 - Post-Hyperparameter Tuning Evaluation and Comparison: Accuracy ISCXURL-2016]



[Figure 35 - Post-Hyperparameter Tuning Confusion Matrices: ISCXURL-2016]

In reference to Table 19, hyperparameter tuning caused RF and GNB to be slightly less performant. RF dropped down to an accuracy of 98.430% from 98.438%, and GNB down to 93.866% from 93.932%. SVM however achieved improvements across all metrics except for a small drop in precision. Its biggest improvement made was its recall which jumped up to 98.369% from 96.508%. Despite this, default RF with feature selection performed the best overall with an accuracy of 98.438%.

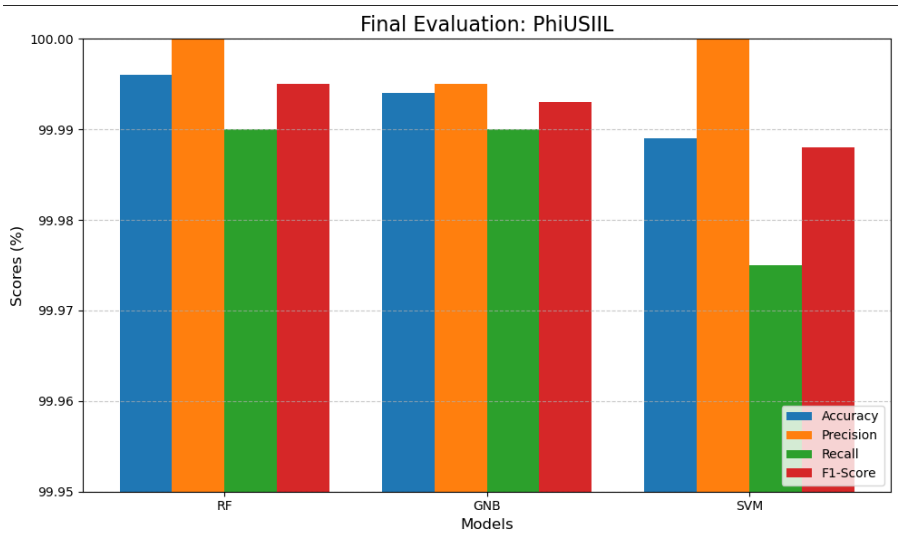
4.5. Final Evaluation

Final evaluation of the optimised models were conducted on the test set in a pseudo 5-fold CV manner. Where the training set was evenly split into 5 subsets, and the same done to the test set. Each model is then trained on each of the 5 training subsets and evaluated on each of the 5 testing subsets.

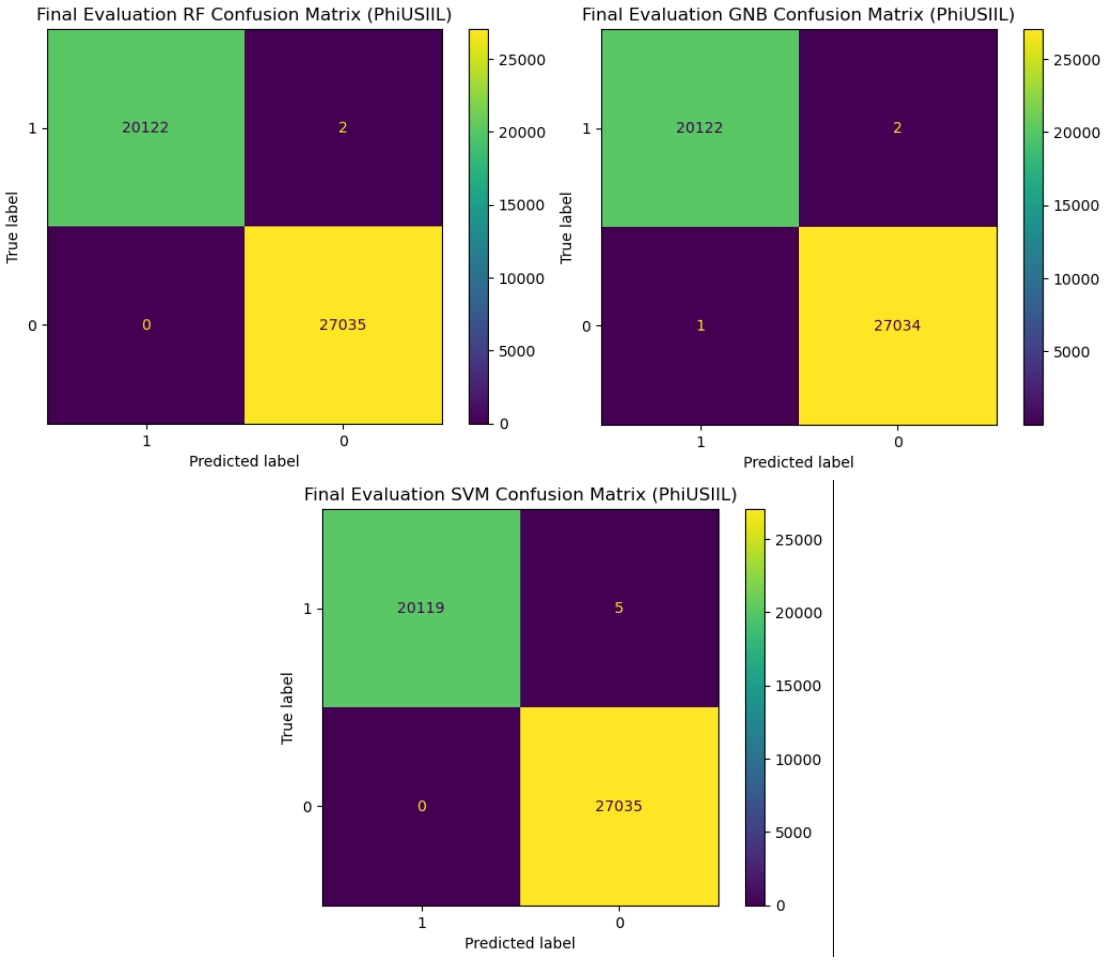
Model	Type	Accuracy %	Precision %	Recall %	F1
PhiUSIIL (USI)	**RF**	**99.996**	**100**	**99.990**	**99.995**
	GNB	99.994	99.995	99.990	99.993
	SVM	99.989	100	99.975	99.988
PhiUSIIL (No USI)	RF	99.741	99.940	99.453	99.696
	GNB	99.308	99.789	98.589	99.185
	SVM	**99.788**	**99.990**	**99.512**	**99.751**
ISCXURL-2016	RF	**96.877**	**97.529**	**96.095**	**96.800**
	GNB	93.527	95.757	90.914	93.269
	SVM	96.096	95.138	97.049	96.080

[Table 20 – Final Evaluations on test set]

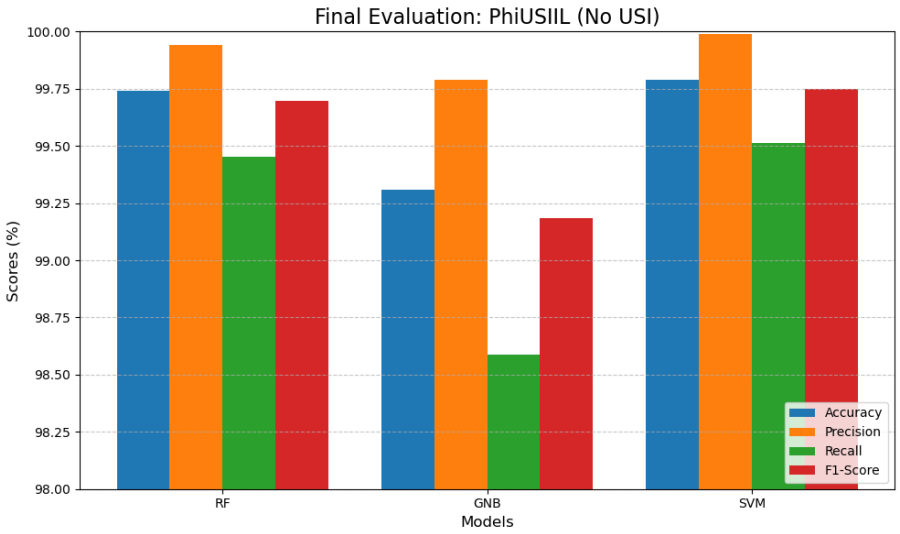
Referencing Table 20, RF is the best overall classifier for PhiUSIIL (USI) with an accuracy of 99.996%, narrowly beating GNB. However, for PhiUSIIL (No USI), SVM performs the best with an accuracy of 99.788%. RF is also the best overall classifier for ISCXURL-2016, achieving an accuracy of 96.877%.



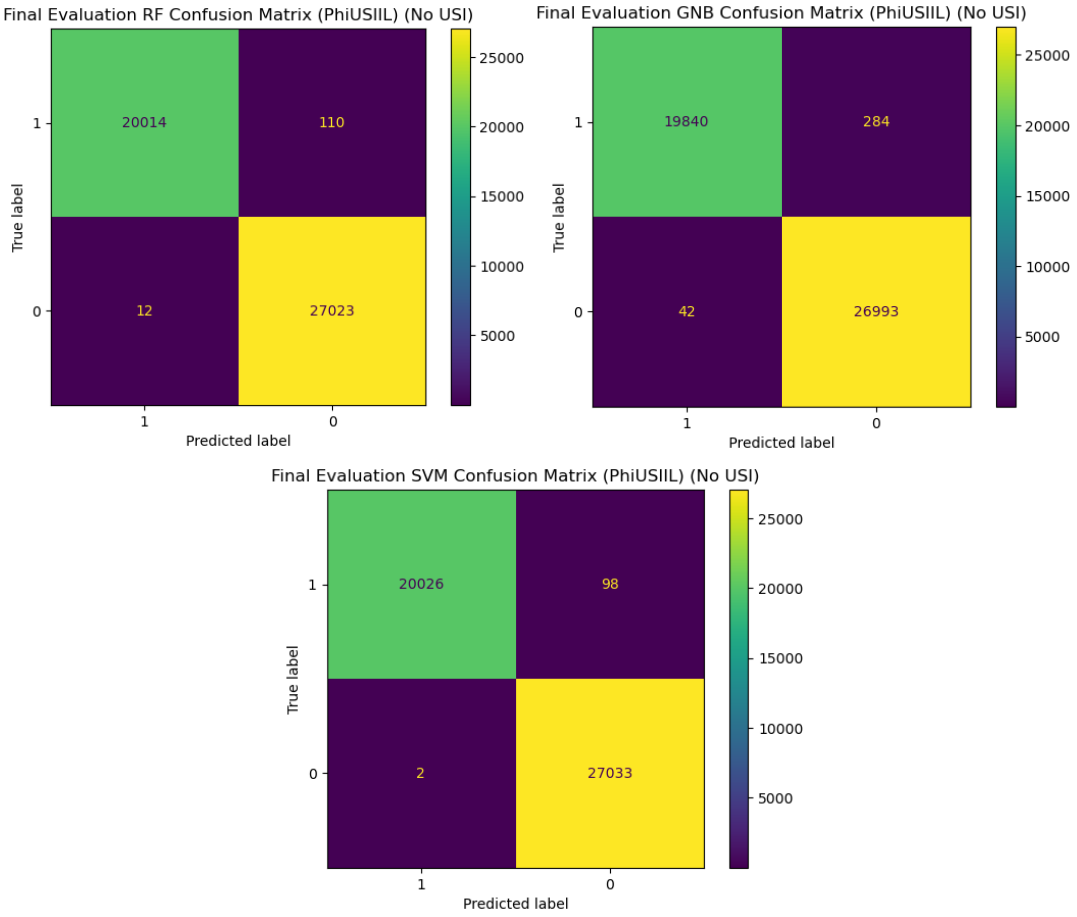
[Figure 36 – Final Evaluation: PhiUSIIL (USI)]



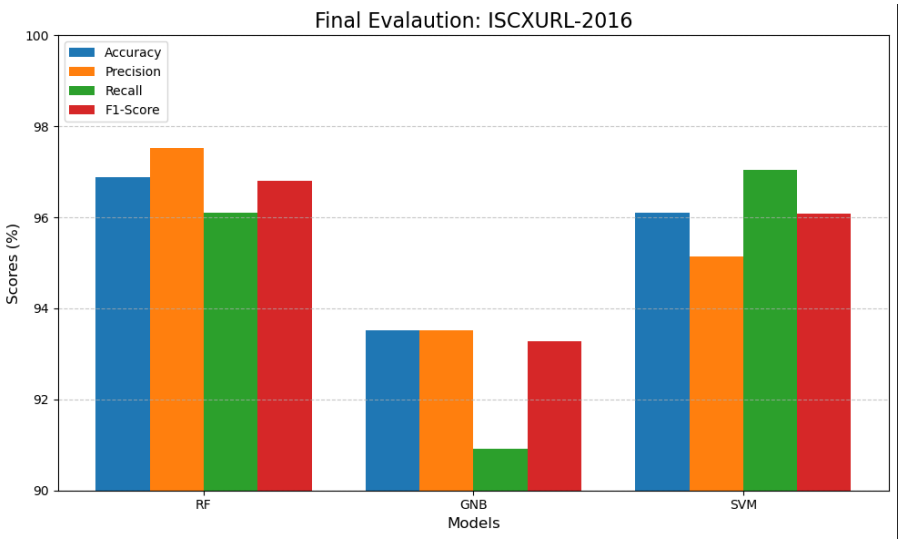
[Figure 37 - Final Evaluation Confusion Matrices: PhiUSIIL (USI)]



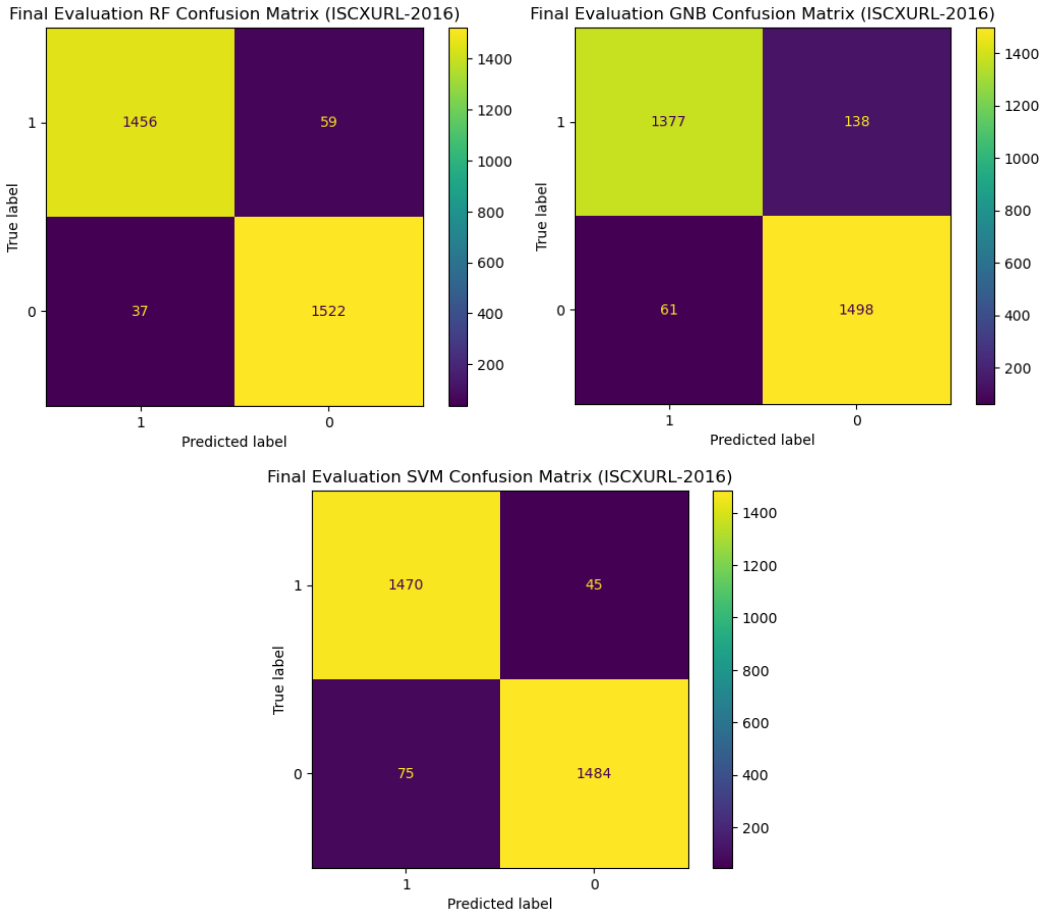
[Figure 38 – Final Evaluation: PhiUSIIL (No USI)]



[Figure 39 - Final Evaluation Confusion Matrices: PhiUSIIL (No USI)]



[Figure 40 – Final Evaluation: ISCXURL-2016]



[Figure 41 - Final Evaluation Confusion Matrices: ISCXURL-2016]

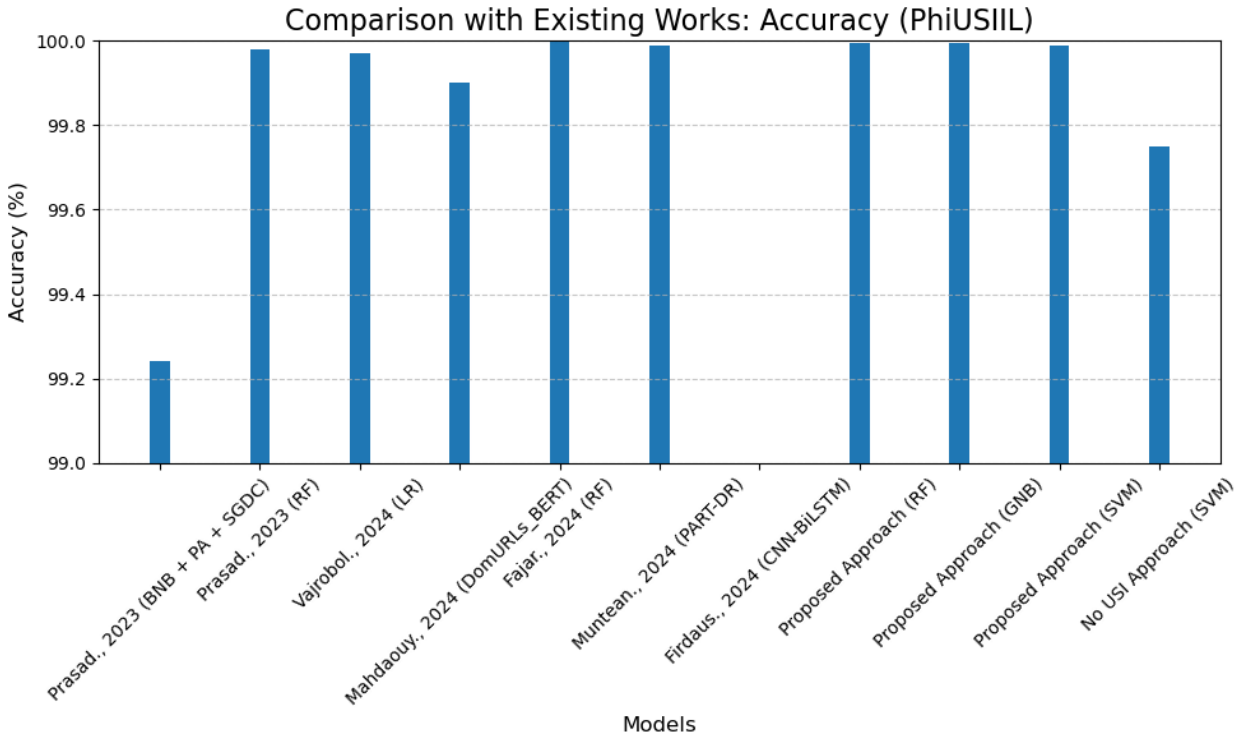
4.6. Comparison to Existing Works

Referencing Table 21 and Figure 42, the proposed RF and GNB models outperform all existing approaches on PhiUSIIl with an accuracy of 99.996% and 99.994% respectively. With the only exception being Fajar et al.'s (2024) approach with RF, CatBoost, and EBM which achieve a perfect accuracy. The proposed SVM also performs extremely well with an accuracy of 99.989% which is only beaten by Fajar et al. (2024) and Muntean (2024). Furthermore, even without the use of USI, the optimised SVM achieves an accuracy of 99.751%, which beats the approaches by Prasad & Chandra (2023) and Firdaus & Sumardi (2024).

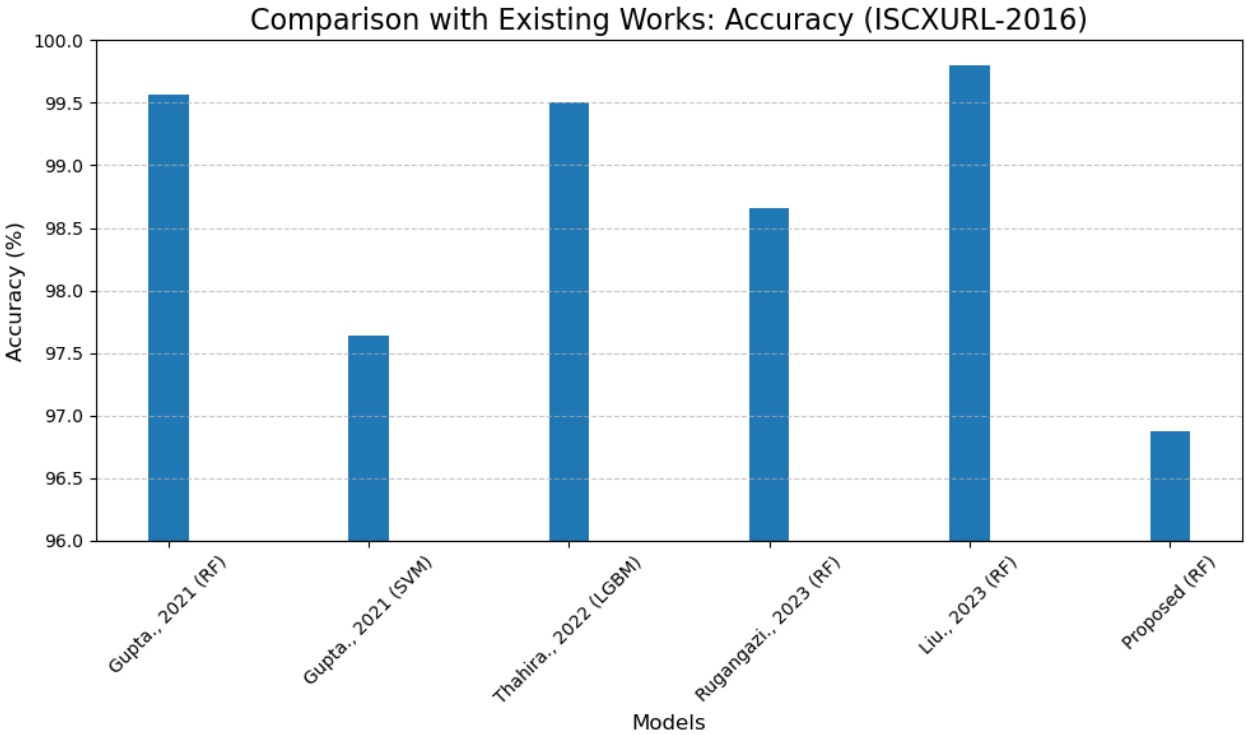
However, on ISCXURL-2016, the proposed approach is outperformed by all existing approaches, as can be seen in Figure 43.

Ref.	Model	Dataset	Accuracy
Prasad & Chandra (2023)	Ensemble (BNB, PA, SGDC), RF	PhiUSIIL	0.9924 (Ensemble), 0.9998 (RF)
Vajrobol et al. (2024)	LR	PhiUSIIL	0.9997
Mahdaouy et al. (2024)	DomURLs_BERT	PhiUSIIL	0.9980
Fajar et al. (2024)	RF, CatBoost, EBM, XGB	PhiUSIIL	1 (RF, CatBoost, EBM), 0.996 (XGB)
Muntean (2024)	PART-DR	PhiUSIIL	0.9999
Firdaus & Sumardi (2024)	Ensemble (CNN-BiLSTM)	PhiUSIIL	0.895
Gupta et al. (2021)	RF, SVM	ISCXURL-2016	0.9957 (RF), 0.9764 (SVM)
Thahira & Ansamma (2022)	LGBM	ISCXURL-2016	0.995
Rugangazi & Okeyo (2023)	RF	ISCXURL-2016	0.9866
Liu et al. (2023)	RF	ISCXURL-2016 (Used all 35,378 URLs)	0.998
Proposed approach	RF	PhiUSIIL, ISCSURL-2016	0.99996 (PhiUSIIL), 0.96877 (ISCSURL-2016)
	GNB	PhiUSIIL	0.99994
	SVM	PhiUSIIL	0.99989
No USI Approach	SVM	PhiUSIIL	0.99751

[Table 21 - Comparison of proposed approach to existing works]



[Figure 42 – Comparison with existing works (PhiUSIIL)]



[Figure 43 – Comparison with existing works (ISCXURL-2016)]

5. Discussion

The research documented in this paper has addressed the research aims and objectives stated in Section 2.6.

Aim 1 has been successfully fulfilled. The performance of RF, GNB, and SVM using lexical URL features for the detection of phishing on the latest URLs has been investigated. Each model has been trained and tested, with metrics documented in tables and illustrated in graphs. As a consequence, RF was found the most performative model out of the tested models consisting of RF, GNB and SVM on both PhiUSIIL and ISCXURL-2016 datasets. RF achieved near perfect accuracy of 99.996% on the PhiUSIIL dataset, which was modified to only include lexical URL features. Additionally, the 9 lexical URL features selected by Gupta et al. (2021) in his work were also extracted and added to the dataset. However, GNB exhibited near perfect accuracy as well, garnering an accuracy of 99.994%, followed by SVM 99.989% accuracy. Decent performance was also achieved by RF on the ISCXURL-2016 dataset, exhibiting an accuracy of 96.877%.

Furthermore, the exceptional performance of default RF on PhiUSIIL during baseline evaluation was investigated. After observing its feature distribution and SHAP value, then performing evaluations after its removal - it was concluded that the performance was mainly attributed to USI. A polarizing feature that distinctly separates target classes in its feature distribution.

In relation to Aim 2, all classifiers have been fine-tuned with the utilisation with one or more optimisation methods consisting of data balancing, feature selection, and hyperparameter tuning. As a result, RF and GNB outperformed all existing works except for Fajar et al.'s (2024). SVM also outperformed a majority of existing works, but also fall short of the work by Fajar et al. (2024), as well as Mundean (2024). While, RF barely improved after feature selection and hyperparameter tuning, GNB and SVM exhibited improvements in performance after feature selection and hyperparameter tuning were performed. Thereby allowing them to outperform approaches which only utlised either feature selection or hyperparameter tuning (Prasad & Chandra, 2023; Vajrobol et al., 2024; Muntean, 2024). It should also be noted that despite this incredible performance, RF, GNB, and SVM also performed the worst out of all selected existing works on the secondary dataset - ISCXURL-2016.

Aim 3 is partially achieved in that some performance evaluations post-data balancing indicated improved performance. However, a majority of the time, balanced data exhibited similar results to that of unbalanced, sometimes even decreasing performance. Hence, the decision to not utilise balanced datasets any further. On top of that, the ISCXURL-2016 dataset was already fairly balanced, along with PhiUSIIL which didn't have a major majority class. However, the imbalance in PhiUSIIL is enough to provide representative conditions likened to that of real life.

Regarding Aim 4, the minimum number of features needed to achieve the same or better performance than that of using all features was found for every model on both dastasets. Additionally, each subset of features contained less features than the number of total features in respective datasets. The largest improvement in performance exhibited by feature selection was observed in the GNB ISCXURL-2016 feature subset, which reduced the number of features from 78 to 39. Consequently, this improved the accuracy of the model from 75.433% to 93.932%. In tandem with the exclusive use of lexical features, using less features after feature selection reduces the computational requirements of the approach (Rugangazi et al., 2023). Therefore, facilitating the possibility of implementing the model locally into a resource-constrained device like a phone or smart watch.

Additionally, feature selection also revealed the limited adaptability of lexical features. To clarify, despite Gupta et al. (2021) being able to achieve 99.57% accuracy using only 9 features on ISCXURL-2016, those same features when adapted into the PhiUSIIL dataset provide varying levels of information and contribution during classification. This is especially so depending on the model. For instance, when MDI feature selection was performed using RF, 'TLDLength' and 'DigitQueryCount' were considered to be amongst the least contributing features. With other extracted features also having low contributions like 'DelimiterDomainCount', 'URLLength', and 'DotCount'. On the other hand, in that same bout of MDI feature selection, 'DelimiterPathCount' and 'LongestPathToken', both extracted features were amongst the top 5 most contributive features.

After feature selection, hyperparameter tuning was performed on each classifier. Consequently, optimal hyperparameters differing from default were found and then evaluated. A majority of the time it resulted in improved performance across accuracy, precision, recall and F1 such as the hyperparameter tuned SVM which exhibited an improved accuracy of 99.992% from 99.987% when compared to the default SVM evaluation on the PhiUSIIL (USI) feature-selected subset. Thereby successfully achieving Aim 5. However, other times it exhibited the exact same performance, like when tuned RF on feature-selected PhiUSILL (USI) achieved the exact same performance as default RF on the same subset. Hyperparameter tuning also saw decreased performance in some case such as tuned GNB on ISCXURL-2016, where the default model on the feature-selected subset returned a higher accuracy of 93.932% compared to the former which achieved 93.866%.

This decrease in performance after hyperparameter tuning may have occurred due to improper testing. Particularly due to hardware limitations, which prevented the use of GridSearchCV to evaluate higher quantities of hyperparameter values and explore more combinations. For instance, values for the n_estimators hyperparameter for RF was never larger than 100 in a grid search as it was too resource intensive and was never able to execute due to the duration of execution. Similarly, values for the C value for SVM were never larger than 10 for the same reason. At the same time, SVM was already an incredibly resource intensive model, especially on PhiUSIIL due to the sheer number of samples.

Consequently, GridSearchCV and RandomisedSearchCV could not be employed due to the extravagant execution durations. Instead, manual tuning using validation curves had to be utilised instead.

Therefore, before pursuing future work like incorporating the optimised RF classifier into a phishing detection framework, evaluating it in real time, and then locally implementing it into a mobile device – hyperparameter tuning should be performed again, using either better hardware or a platform like Google Colab. So that varying values of hyperparameters and more combinations can be explored. Particularly for models on datasets where they exhibited decreased performance afterwards.

6. Conclusion

Phishing attacks are a significant ongoing cybersecurity challenge as a result of their low complexity, high effectiveness, and ability to exploit human vulnerabilities. This research has demonstrated the potential of utilising lexical URL features in combination with fine-tuned machine learning approaches to address these challenges effectively.

The study successfully showcased the efficacy of RF, GNB, and SVM classifiers on the recent dataset PhiUSIIL which focuses on the recency of URLs. To which, the RF and GNB classifiers achieved near-perfect accuracy. Outperforming a large majority of existing works evaluated on the dataset, The efficacy of these classifiers were also examined on an older dataset, ISCXURL-2016 which has been used numerously in phishing detection research. However, the classifiers found the classification task on this dataset challenging in comparison.

In response to the lack of utilisation of optimisation techniques such as data balancing, feature selection and hyperparameter tuning, testing was conducted to explore their efficacy. Consequently, it was found that model performance generally improved after feature selection and hyperparameter tuning. However, data balancing mainly exhibited similar performance to unbalanced data, and was therefore excluded from further analysis. Despite this, this research still highlights the importance of fine-tuning models rather than solely focusing on complex innovative algorithms and approaches.

The importance of the extremely polarizing feature ‘URLSimilarityIndex’ was also explored, which explained the near-perfect performance of the RF classifier on PhiUSIIL.

Despite its successes, the study acknowledges certain limitations, such as the hardware in which the implementation was conducted on. This affects aspects of the experiment, such as the inability to use 10-fold cross-validation, certain values for model hyperparameters, and automatic methods of hyperparameter tuning for SVM.

By focusing on lexical URL features, the proposed models are language-independent and capable of addressing zero-day phishing attacks. A future implementation as a locally integrated mobile application could provide real-time protection against phishing without relying on external APIs, mitigating risks of downtime or denial-of-service attacks. Such advancements could greatly reduce the prevalence of phishing, protecting individuals and organizations from identity theft, financial losses, and broader cyber threats. All without the user having to have any knowledge about cybersecurity.

This research contributes to the field of machine learning phishing detection by emphasizing the utilisation of feature selection and hyperparameter tuning. It also, provides greater insight into the remarkable feature that is ‘URLSimilarityIndex’ of the PhiUSIIL dataset. The findings serve as a foundation for future developments, encouraging the continued integration of optimized models into real-world applications to foster a safer digital environment for users.

Acknowledgements

I would like to thank my supervisor Dr. Kun Yu for his ongoing support and insight throughout the entire lifespan of this project.

References

Abdul Samad, S. R., Balasubaramanian, S., Al-Kaabi, A. S., Sharma, B., Chowdhury, S., Mehbodniya, A., Webber, J. L., & Bostani, A. (2023). Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection. *Electronics*, *12*(7), 1642. https://doi.org/10.3390/electronics12071642

Afroz, S., & Greenstadt, R. (2011, 2011). PhishZoo: Detecting Phishing Websites by Looking at Them.

Aggrawal, R., & Pal, S. (2020). Sequential feature selection and machine learning algorithm-based patient's death events prediction and diagnosis in heart disease. *SN Computer Science*, *1*(6), 344.

APWG. (2023). *APWG Phishing Attack Trends Report - 4Q 2023*. APWG. https://apwg.org/trendsreports/

APWG. (2024). *APWG Phishing Attack Trends Report - 1Q 2024*. APWG. https://apwg.org/trendsreports/

Awasthi, A., & Goel, N. (2022). Phishing website prediction using base and ensemble classifier techniques with cross-validation. *Cybersecurity*, *5*(1). https://doi.org/10.1186/s42400-022-00126-9

Basit, A., Zafar, M., Liu, X., Javed, A. R., Jalil, Z., & Kifayat, K. (2021). A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommunication Systems*, *76*(1), 139-154. https://doi.org/10.1007/s11235-020-00733-2

Cinar, A. C., & Kara, T. B. (2023). The current state and future of mobile security in the light of the recent mobile security threat reports. *Multimedia Tools and Applications*, *82*(13), 20269-20281. https://doi.org/10.1007/s11042-023-14400-6

CrowdStrike. (2024). *CrowdStrike 2024 Global Threat Report*. CrowdStrike. https://www.crowdstrike.com/global-threat-report/

Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions. *IEEE Access*, *10*, 36429-36463. https://doi.org/10.1109/access.2022.3151903

Fajar, A., Yazid, S., & Budi, I. (2024). Enhancing Phishing Detection through Feature Importance Analysis and Explainable AI: A Comparative Study of CatBoost, XGBoost, and EBM Models. *arXiv preprint arXiv:2411.06860*.

Firdaus, D., & Sumardi, I. (2024). Phishing Website Detection Using Ensemble Algorithm Convolutional Neural Network and Bidirectional LSTM. *Industrial Sciencetech Journal*, *1*(1), 14-23.

Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Computer Communications*, *175*, 47-57. https://doi.org/10.1016/j.comcom.2021.04.023

Han, H., Guo, X., & Yu, H. (2016). Variable selection using mean decrease accuracy and mean decrease gini based on random forest. 2016 7th ieee international conference on software engineering and service science (icsess),

Hillman, D., Harel, Y., & Toch, E. (2023). Evaluating organizational phishing awareness training on an enterprise scale. *Computers & Security*, *132*, 103364. https://doi.org/10.1016/j.cose.2023.103364

Jalil, S., Usman, M., & Fong, A. (2023). Highly accurate phishing URL detection based on machine learning. *Journal of Ambient Intelligence and Humanized Computing*, *14*(7), 9233-9251. https://doi.org/10.1007/s12652-022-04426-3

Limna, P., Kraiwanit, T., & Siripipattanakul, S. (2023). The Relationship between Cyber Security Knowledge, Awareness and Behavioural Choice Protection among Mobile Banking Users in Thailand. *International Journal of Computing Sciences Research*, 1133-1151%V 1137. //stepacademic.net/ijcsr/article/view/378

Liu, S., Wu, H., Cheng, G., & Hu, X. (2023). Real-Time Phishing Detection Based on URL Multi-Perspective Features: Aiming at the Real Web Environment. ICC 2023-IEEE International Conference on Communications,

Mahdaouy, A. E., Lamsiyah, S., Idrissi, M. J., Alami, H., Yartaoui, Z., & Berrada, I. (2024). DomURLs_BERT: Pre-trained BERT-based Model for Malicious Domains and URLs Detection and Classification. *arXiv preprint arXiv:2409.09143*.

Mamun, M. S. I., Rathore, M. A., Lashkari, A. H., Stakhanova, N., & Ghorbani, A. A. (2016). Detecting Malicious URLs Using Lexical Analysis. In (pp. 467-482). Springer International Publishing. https://doi.org/10.1007/978-3-319-46298-1_30

Mishra, S., & Soni, D. (2019, 2019). SMS Phishing and Mitigation Approaches.

Muntean, M. V. (2024). Preprocessing methods for improving phishing URL detection. 2024 International Conference on INnovations in Intelligent SysTems and Applications (INISTA),

Prasad, A., & Chandra, S. (2023). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. *Computers & Security*, *136*, 103545. https://doi.org/10.1016/j.cose.2023.103545

Rugangazi, B., & Okeyo, G. (2023). Detecting Phishing Attacks Using Feature Importance-Based Machine Learning Approach. 2023 IEEE AFRICON,

Sáez-De-Cámara, X., Flores, J. L., Arellano, C., Urbieta, A., & Zurutuza, U. (2023). Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks. *Computers & Security*, *131*, 103299. https://doi.org/10.1016/j.cose.2023.103299

Thahira, A., & Ansamma, J. (2022, 2022). Phishing Website Detection Using LGBM Classifier With URL-Based Lexical Features.

Vajrobol, V., Gupta, B. B., & Gaurav, A. (2024). Mutual information based logistic regression for phishing URL detection. *Cyber Security and Applications*, *2*, 100044. https://doi.org/10.1016/j.csa.2024.100044

Verizon. (2024). *Verizon 2024 Data Breach Investigations Report*. Verizon. https://www.verizon.com/business/en-au/resources/reports/dbir/

Zheng, M., Wang, F., Hu, X., Miao, Y., Cao, H., & Tang, M. (2022). A Method for Analyzing the Performance Impact of Imbalanced Binary Data on Machine Learning Models. *Axioms*, *11*(11), 607. https://doi.org/10.3390/axioms11110607

Appendix

Github page link to research code and datasets: https://github.com/Riddle920/Honours/tree/main

(How ironic :P)

ASSIGNMENT COVERSHEET

UTS: ENGINEERING & INFORMATION TECHNOLOGY

SUBJECT NUMBER & NAME	NAME OF STUDENT(s) (PRINT CLEARLY)	STUDENT ID(s)
31482 – Honours Project	Hugh Riddle *SURNAME* *FIRST NAME*	14241323

STUDENT EMAIL	STUDENT CONTACT NUMBER
Hugh.riddle@student.uts.edu.au	

NAME OF TUTOR	TUTORIAL GROUP	DUE DATE
Dr. Kun Yu		22/11/24

ASSESSMENT ITEM NUMBER & TITLE

Assessment Task 2: Research Report and Research Work

I confirm that I have read, understood and followed the guidelines for assignment submission and presentation on page 2 of this cover sheet.

I confirm that I have read, understood and followed the advice in the Subject Outline about assessment requirements.

I understand that if this assignment is submitted after the due date it may incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Declaration of originality: The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s) and has not been previously submitted for assessment. I understand that, should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with University policy and rules. In the statement below, I have indicated the extent to which I have collaborated with others, whom I have named.

Statement of collaboration:

Signature of student(s): Hugh Riddle Date: 22/11/2024

"- -- -

ASSIGNMENT RECEIPT

To be completed by the student if a receipt is required

SUBJECT NUMBER & NAME	NAME OF TUTOR	
SIGNATURE OF TUTOR		RECEIVED DATE