International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

SmartShield NLP: Context-Aware Threat Severity Prediction System

Harsh Sadariya[1], Dr. Mohit Bhadla[2]

[1]Master Student's, [2]Associate Professor
[1,2]Department of Computer Science and Engineering, Gandhinagar Institute of Technology, Kalol, Gujarat, India

Abstract:
Traditionally the binary classification models used by cybersecurity systems merely describe threats as either malicious or benign. Although useful at the early stages of filtering, such binary methods do not necessarily provide the picture of the context and seriousness of the threats, and thus it becomes difficult to prioritize incident response and resource allocation. Phishing in the modern shifting threat environment is one of the most prevalent and successful areas of cyber intrusion including both simple and innocent spam and much more serious and malicious attacks with an objective to commit either credentials theft or to deploy ransomware. To address this deficiency, this study seeks to present Beyond Binary: NLP-Based Threat Severity Prediction to Enhanced Security Response that uses Natural Language Processing (NLP) to scan the phishing emails and subsequently categorizes the emails based on their respective levels of threat rather than a simple safe/unsafe result.
Phishing email datasets (enron and phish tank) are used as experimental data set in the study; the datasets are a rich source of real-life textual data. The emails are preprocessed using Python and Google Colab as a development environment through tokenization, removal of stopwords, lemmatization and embedding which are implemented through TF-IDF and transformer-based embeddings (BERT). Next, machine learning and deep learning models (Random Forest, XGBoost, LSTM and Transformer architectures) are trained to predict the levels of severity in the categories of low, medium, high and critical. This multi-class severity prediction goes beyond the binary system of phishing detection, providing high-resolution information about the possible effects of the threat.
The suggested framework enhances security response mechanisms due to the fact that it enables prioritization of threats in addition to intelligent management of alerts. Threats of low severity can be automatically filtered, medium-level phishing tackles limited to review and high-to-critical level threats sent to the incident response teams to act immediately. This results in an increased usage of resources and reduces the problem of alert fatigue within Security Operations Centers (SOCs). This study is expected to produce a more scalable NLP-driven model, with the capacity to detect the existence of phishing attacks and respond to them in a context-aware and severity-based manner to offer a higher level of resilience to the current cyberattacks.

Keywords: NLP, Phishing Emails, Threat Severity Prediction, Cybersecurity, Machine Learning.

I. INTRODUCTION

Cybersecurity remains a major concern in the digital era, with phishing attacks being one of the most common and effective methods of intrusion [1]. Traditional detection methods such as signature-based systems, blacklisting, and rule-based filters offer basic protection, but attackers continuously evolve their tactics to bypass them [2]. As a result, machine learning and Natural Language Processing (NLP) have emerged as powerful tools for analyzing unstructured data like emails and threat reports [3].
Phishing emails are particularly challenging due to their text-based, dynamic, and context-dependent nature [4]. Most existing systems use binary classification to label emails as phishing or non-phishing,

which improves detection accuracy but fails to assess the severity of threats [5]. This means low-risk spam and high-risk phishing attacks are treated equally, leading to alert fatigue and delayed response to critical threats [5].

Recent research emphasizes the need for multi-class classification systems that can evaluate both the presence and severity of threats [6]. Advances in NLP, including word embeddings (Word2Vec, GloVe), contextual models like BERT and GPT, and hybrid deep learning approaches, enable better understanding of linguistic and semantic patterns [7]. Integrating these models into systems like SIEM and SOAR allows organizations to prioritize high-risk threats effectively [8].

This study builds on existing phishing detection research by proposing an NLP-based framework for severity prediction, aiming to improve threat prioritization and real-world response efficiency.

A. Scope of the Study

This study focuses on developing a Natural Language Processing (NLP)-based system to classify phishing emails beyond simple binary detection. Instead of only identifying whether an email is phishing or not, it aims to predict the severity level—low, medium, high, and critical—providing a more detailed understanding of threats and enabling better prioritization of responses. The research uses a moderately sized dataset by combining Enron emails and PhishTank data, ensuring it remains manageable while still reflecting real-world phishing scenarios.

The methodology involves multiple stages, including text preprocessing using tokenization, stopword removal, and lemmatization, followed by feature extraction techniques such as TF-IDF, word embeddings, and transformer-based embeddings. Various models are evaluated, including traditional machine learning approaches like Random Forest and XGBoost, as well as deep learning models such as LSTM and BERT, to improve the accuracy of severity prediction.

The scope is limited to predicting phishing email severity and does not cover areas like incident response, malware analysis, or real-time deployment. However, the results can be integrated into Security Operations Center (SOC) systems such as SIEM and SOAR to enhance alert prioritization and workflow efficiency. This approach supports context-aware cybersecurity systems, helps analysts make better decisions, and improves organizational resilience by enabling faster responses to high-risk phishing threats.

B. Research Objectives

The primary objective of this study is to enhance existing phishing detection mechanisms by moving beyond binary classification toward a multi-level severity prediction approach. The research aims to develop a model that can accurately classify phishing emails into different severity levels, thereby providing a more detailed understanding of threats and overcoming the limitations of traditional detection systems.

Another objective is to evaluate the effectiveness of severity-based classification in reducing alert fatigue and improving prioritization within a Security Operations Center (SOC). The study seeks to analyze how multi-level severity predictions can help security analysts focus on high-risk threats more efficiently compared to conventional binary systems.

The study also aims to design and assess a scalable NLP-based severity prediction model that can be integrated into real-world SOC environments. This includes exploring how such a system can support automated responses, enhance decision-making processes, and improve overall operational efficiency in cybersecurity workflows.

C. Research Gaps

Despite advancements in phishing detection, several critical gaps remain that limit real-world effectiveness. Most existing systems rely on binary classification, distinguishing only between phishing and legitimate emails without considering severity or potential impact. This lack of granularity treats all threats equally—from low-risk spam to high-risk attacks—leading to alert fatigue, poor prioritization,

and delayed response in Security Operations Centers (SOCs). Additionally, many models depend heavily on surface-level features such as URLs, metadata, and lexical patterns, while underutilizing deeper contextual and semantic understanding. Advanced NLP techniques like transformer-based models (e.g., BERT) are still rarely applied for interpreting threat severity.

Another major limitation is the lack of publicly available datasets with severity labels, which are essential for training multi-class models capable of predicting low, medium, high, and critical threat levels. Most datasets are binary-labeled, restricting the development of severity-aware systems. Furthermore, existing research often lacks comprehensive comparisons between traditional machine learning and modern deep learning models for multi-class severity prediction. Studies typically focus on a single approach and evaluate only accuracy, without analyzing performance across different severity levels.

Finally, there is a significant gap between academic research and real-world SOC implementation. While phishing detection techniques are widely studied, few works demonstrate how severity-based predictions can improve alert prioritization, reduce analyst workload, or enhance automated responses. This disconnect highlights the need for practical, scalable solutions that bridge theoretical advancements with operational cybersecurity requirements.

II. LITERATURE REVIEW

Çelik et al. (2025) propose an NLP-based framework to enhance phishing detection in financial systems using TF-IDF, clustering, and semantic similarity techniques. Their approach utilizes the Universal Sentence Encoder to capture contextual meaning in emails, enabling better identification of unseen phishing patterns. The model significantly improves detection accuracy compared to traditional blacklist systems and reduces false negatives in financial datasets. However, the system is limited to binary classification and cannot differentiate between low-risk and high-risk phishing emails. This restricts its ability to support threat prioritization in real-world security environments [11].

Sallouma et al. (2021) explore phishing detection using machine learning models that combine lexical, host-based, and content-based features. The study demonstrates that ensemble models such as Random Forest and Gradient Boosting improve classification accuracy and reduce false positives. By integrating multiple feature types, the system captures diverse phishing indicators more effectively than single-feature approaches. However, the research primarily focuses on phishing websites rather than email content analysis. Additionally, it relies on binary classification, limiting its ability to assess the severity or urgency of threats [12].

Dey et al. (2023) introduce a deep learning-based phishing detection system using CNN and RNN architectures to analyze URLs and webpage content. The hybrid approach captures both spatial and sequential patterns, allowing the model to learn complex phishing behaviors. Experimental results show that the model outperforms traditional machine learning techniques in accuracy and robustness. However, the system requires high computational resources, making deployment challenging in resource-constrained environments. Furthermore, it does not address email-based phishing detection or severity classification [13].

Mittal and Engels (2022) develop a hybrid phishing detection model that combines machine learning and deep learning techniques, including ANN, CNN, RNN, XGBoost, and word embeddings. This integrated approach enables the system to capture linguistic, semantic, and contextual features of phishing emails effectively. The model demonstrates improved accuracy and robustness compared to single-model approaches. However, the complexity of combining multiple algorithms results in high computational overhead. Additionally, the system is limited to binary classification and does not provide severity-based threat analysis [14].

Kavya and Sumathi (2025) review recent advancements in phishing detection, focusing on hybrid deep learning models such as ResNeXt-GRU and GAN-based systems. These models achieve high accuracy levels, often exceeding 98%, and improve detection of evolving phishing techniques. The use of GANs

also helps generate synthetic phishing samples for better training. However, these advanced models require significant computational resources, making real-time deployment difficult. Moreover, the study does not address severity-based classification, limiting its practical use in threat prioritization [15].

Khan et al. (2025) propose an AI-driven cybersecurity framework combining Artificial Neural Networks with Information Security Management (ANN-ISM). The model enhances predictive capabilities and supports proactive threat detection through continuous monitoring. Experimental results indicate improved early warning systems and reduced false positives. However, the framework does not specifically target phishing detection or analyze email content using NLP techniques. Additionally, it lacks multi-level severity classification, limiting its effectiveness in prioritizing threats [16].

Mladenovic et al. (2024) apply NLP-based sentiment analysis and optimized machine learning models to detect insider threats. The study uses techniques such as TF-IDF and metaheuristic optimization to improve classification performance. Results show increased accuracy and reduced training time due to optimization methods. However, the research is limited to insider threats and does not consider phishing attacks or external cyber threats. It also lacks multi-class severity classification, reducing its applicability in broader cybersecurity contexts [17].

Salem et al. (2024) provide a comprehensive review of machine learning, deep learning, and metaheuristic techniques in cybersecurity. The study highlights that hybrid models outperform single-method approaches in detecting threats and improving accuracy. It also emphasizes the role of optimization algorithms in enhancing model performance. However, the review is largely theoretical and does not focus on phishing email detection specifically. Additionally, it does not explore severity-based classification, leaving a gap in fine-grained threat analysis [18].

Ferrag et al. (2024) examine the role of federated learning in cybersecurity, enabling collaborative model training without sharing sensitive data. The approach improves data privacy and allows organizations to share threat intelligence securely. Experimental results show comparable accuracy to centralized models. However, challenges such as communication overhead and vulnerability to attacks remain. Furthermore, the study does not address phishing email detection or severity-based classification, limiting its applicability [19].

Silvestri et al. (2024) investigate adversarial machine learning techniques and their impact on cybersecurity systems. The study proposes defense mechanisms such as adversarial training and ensemble methods to improve model robustness. Results show increased resilience against adversarial attacks. However, the research focuses on general threat detection and does not specifically address phishing email analysis. It also lacks multi-level severity classification, limiting its usefulness for threat prioritization [20].

Admass et al. (2024) provide a comprehensive review of cybersecurity advancements, emphasizing the role of artificial intelligence, machine learning, and NLP in modern threat detection systems. The study highlights how automation and intelligent systems improve detection accuracy and response efficiency. It also discusses challenges such as scalability, adversarial attacks, and integration with human decision-making. However, the research remains conceptual and does not focus specifically on phishing email detection. Additionally, it lacks discussion on multi-level severity classification, limiting its applicability to threat prioritization systems [1].

Al-Subaiey et al. (2024) develop an interpretable AI-based platform for phishing website detection, focusing on transparency and explainability in decision-making. The system uses interpretable machine learning models to help analysts understand why a website is classified as phishing. Results show improved trust and usability without compromising accuracy. However, the study is limited to website-based phishing detection and does not analyze email content. It also does not support multi-class severity classification, restricting its use in advanced phishing analysis [4].

Orunsolu et al. (2022) propose a predictive phishing detection model using machine learning techniques based on URL and webpage features. The model achieves higher accuracy compared to traditional blacklist systems by leveraging structural and heuristic features. It is effective in identifying phishing

patterns in real-time environments. However, the approach lacks NLP-based semantic analysis and does not consider email content. Additionally, it only supports binary classification and does not provide severity-level predictions [5].

Alaeifar et al. (2024) explore NLP-based cyber threat intelligence extraction to improve information sharing among organizations. The study demonstrates how techniques like named entity recognition and semantic analysis can extract meaningful insights from unstructured threat data. This enhances collaboration and situational awareness across cybersecurity systems. However, the research does not specifically focus on phishing detection or email-based threats. It also lacks severity classification mechanisms, limiting its use in prioritizing cyber threats [6].

Naz et al. (2025) investigate machine learning and deep learning techniques for personality trait detection using textual data. The study demonstrates the effectiveness of NLP models such as TF-IDF, CNN, and LSTM in capturing semantic and contextual patterns. Results show improved accuracy in identifying psychological traits from text. However, the research is not related to cybersecurity or phishing detection. It does not address malicious intent detection or severity classification, limiting its relevance to phishing research [7].

Tanti (2024) provides an overview of phishing attacks, their types, and prevention techniques. The study categorizes phishing methods such as spear phishing, clone phishing, and whaling, offering insights into attacker strategies. It also highlights preventive measures like user awareness and authentication techniques. However, the research is descriptive and lacks technical depth in machine learning or NLP-based detection methods. It does not include experimental validation or severity-based classification approaches [9].

Kearney et al. (2025) analyze the issue of alert fatigue in Security Operations Centers, where analysts are overwhelmed by large volumes of alerts. The study identifies binary detection systems as a major cause due to their inability to differentiate threat severity. It emphasizes the need for intelligent prioritization mechanisms to improve response efficiency. Results show that severity-based alerting can significantly reduce workload and improve decision-making. However, the study does not provide a technical implementation or NLP-based model for severity prediction [10].

Sallouma et al. (2021) present a survey of NLP-based phishing email detection techniques, highlighting the importance of linguistic features such as urgency and deception. The study compares traditional machine learning models with deep learning approaches, showing improved accuracy with advanced models. It emphasizes the role of semantic and contextual understanding in phishing detection. However, the survey focuses on binary classification and does not explore multi-level severity prediction. It also lacks practical implementation and experimental validation [12].

Safi and Singh (2023) conduct a systematic review of phishing website detection techniques, comparing heuristic, machine learning, and deep learning approaches. The study finds that hybrid models outperform traditional methods in terms of accuracy and robustness. It also highlights challenges such as dataset imbalance and adversarial attacks. However, the research is limited to website-based phishing detection and does not address email content analysis. Additionally, it does not consider severity-based classification, leaving a gap in threat prioritization [21].

Li et al. (2024) review state-of-the-art deep learning models for phishing detection, including CNN, RNN, and hybrid architectures. The study demonstrates that deep learning models achieve higher accuracy by capturing complex patterns in phishing data. It also highlights improvements in scalability and generalization. However, the research focuses on website phishing and does not incorporate NLP-based email analysis. Furthermore, it does not explore multi-class severity classification, limiting its relevance to advanced phishing detection systems [22].

III. PROPOSED FLOW OF THE RESEARCH

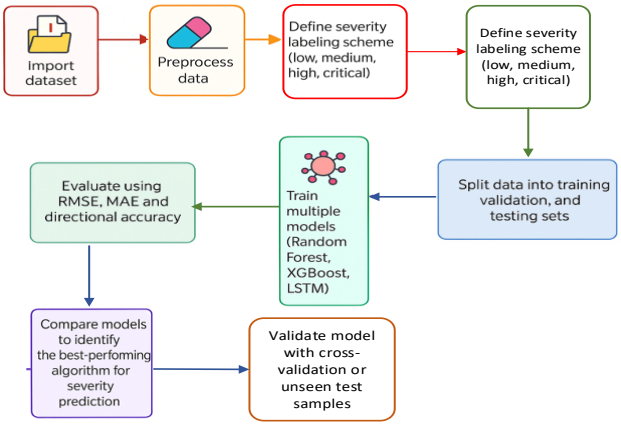

Fig 1: Proposed Flow of the Research

This study uses two major datasets: the Enron Email Dataset, which provides real-world email content including message body, subject, sender, recipient, and metadata, and the PhishTank Dataset, which contains verified phishing URLs along with details such as target brand and verification status. By combining these sources, the study integrates both raw email content and phishing-specific metadata. Additional derived features such as keyword presence, URL indicators, target criticality, and verification status are used to assign multi-class severity labels—low, medium, high, and critical—creating a more comprehensive dataset.

The methodology begins with data preprocessing, where emails are cleaned by removing noise, duplicates, and irrelevant characters. NLP techniques such as tokenization, stopword removal, lemmatization, and part-of-speech tagging are applied to standardize the text. Extracted URLs are validated against the PhishTank dataset to confirm phishing instances, and features like suspicious keyword flags and verified phishing indicators are generated. Feature extraction combines statistical methods like TF-IDF with semantic techniques such as Word2Vec, GloVe, and transformer-based embeddings like BERT to capture both surface-level and deep contextual information from the emails.

In the modeling phase, the dataset is split into training and testing sets, and both traditional machine learning models (Logistic Regression, Random Forest, XGBoost) and deep learning models (LSTM and BERT) are trained to classify emails into four severity levels. Model performance is evaluated using cross-validation and metrics such as accuracy, precision, recall, F1-score, and confusion matrices. Finally, the study proposes integrating the severity-based system into Security Operations Center (SOC) workflows, where low-risk emails are filtered automatically, medium-risk emails are reviewed by analysts, and high or critical threats are escalated immediately. This approach improves prioritization, reduces alert fatigue, and demonstrates the practical value of severity-aware phishing detection.

IV. IMPLEMENTATION

A. Dataset Overview

This study utilizes two complementary datasets to support phishing detection and severity classification. The Email.csv dataset contains semi-structured email data where each record represents a complete email including metadata such as message ID, sender, receiver, subject, and date, along with the message body. While metadata provides communication context, the email body contains unstructured natural language text, which is essential for tasks like phishing detection and text classification. During preprocessing, irrelevant elements such as system headers and transmission-related data are removed to reduce noise and improve model efficiency.

The second dataset is a structured phishing dataset containing verified phishing records with attributes such as URL, submission time, verification time, target brand, and verification status. These features

help analyze phishing behavior, attack persistence, and target trends. By combining email content with phishing metadata, the study enables a comprehensive analysis of both linguistic and contextual indicators of phishing attacks.

B. Data Cleaning & Feature Engineering

- **Library Import and Dependencies**: The implementation uses multiple Python libraries to support the machine learning pipeline. Pandas is used for data handling and preprocessing, while regular expressions (re) help clean textual data by removing noise and unwanted characters. TF-IDF Vectorizer converts text into numerical features, enabling machine learning models to process textual input. Matplotlib is used for visualization of results, while train-test split ensures proper evaluation of the model on unseen data. The Random Forest classifier is selected due to its robustness and ability to handle high-dimensional data. Evaluation metrics such as accuracy, confusion matrix, precision, recall, and F1-score are used to measure model performance effectively.
- **Dataset Downloading and Loading**: Datasets are obtained from external sources and loaded into the working environment for processing. The Enron Email Dataset is downloaded using the Kaggle API, while the phishing dataset is loaded from a CSV file. After loading, the email dataset is sampled to 50,000 records to reduce computational complexity, and the phishing dataset contains 7,411 records. These datasets are structured into Pandas DataFrames, ensuring consistency and readiness for further analysis. This step establishes a strong foundation for subsequent preprocessing and modeling.

C. Message Cleaning and Text Extraction

Message Cleaning and Text Extraction

Raw email data contains both metadata and content, but only the message body is relevant for phishing detection.

```
def extract_body_enron(message):
    if not isinstance(message, str):
        return ""

    parts = re.split(r"\n\s*\n", message, maxsplit=1)
    body = parts[1] if len(parts) > 1 else message

    reply_patterns = [
        r"\nFrom:.*",
        r"\nTo:.*",
        r"\nCc:.*",
        r"\nSubject:.*",
        r"\n-----Original Message-----.*",
        r"\nOn .* wrote:.*"
    ]

    for pattern in reply_patterns:
        body = re.split(pattern, body, maxsplit=1, flags=re.IGNORECASE)[0]

    body = re.sub(r"^[=\-_*]{3,}$", "", body, flags=re.MULTILINE)

    body = re.sub(r"\n{2,}", "\n", body)

    return body.strip()
df["body"] = df["message"].apply(extract_body_enron)

print("ORIGINAL MESSAGE:\n")
print(df["message"].iloc[1])

print("\n" + "="*50 + "\n")

print("EXTRACTED BODY:\n")
print(df["body"].iloc[1])
```

Fig 2: Email Body Extraction and Cleaning using Custom Python Function

Therefore, preprocessing focuses on extracting meaningful text while removing headers, forwarded content, and redundant information. A custom Python function is used to automate this process, ensuring consistent cleaning across all emails.

```
ORIGINAL MESSAGE:

Message-ID: <22688499.1075854130303.JavaMail.evans@thyme>
Date: Mon, 24 Apr 2000 05:43:00 -0700 (PDT)
From: pat.clynes@enron.com
To: aimee.lannou@enron.com
Subject: Meter #1591 Lamay Gaslift
Cc: daren.farmer@enron.com
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Bcc: daren.farmer@enron.com
X-From: Pat Clynes
X-To: Aimee Lannou
X-cc: Daren J Farmer
X-bcc:
X-Folder: \Darren_Farmer_Dec2000\Notes Folders\Logistics
X-Origin: Farmer-D
X-FileName: dfarmer.nsf

Aimee,
Please check meter #1591 Lamay gas lift.  It doesn't appear to have very much
flow and the
BAV is showing the nom volume.  This could be adversely affecting the risk
numbers.  Pat

------------------------------------------------

EXTRACTED BODY:

Aimee,
Please check meter #1591 Lamay gas lift.  It doesn't appear to have very much
flow and the
BAV is showing the nom volume.  This could be adversely affecting the risk
numbers.  Pat
```

Fig 3: Output of Email Body Extraction Before and After Cleaning

Additional steps include removing special characters, duplicate content, and formatting inconsistencies, resulting in clean and standardized text suitable for NLP tasks.

URL Extraction and Processing

URLs play a crucial role in phishing detection, as malicious links are commonly embedded in phishing emails. Regular expressions are used to extract URLs from email content. Extracted URLs are stored separately, counted, and removed from the text to avoid interference with linguistic analysis. New features such as URL count and extracted links are generated, enabling the model to analyze both textual patterns and link-based indicators. This enhances the overall effectiveness of the detection system.

D. Text Vectorization

Text Preprocessing and TF-IDF Vectorization

After cleaning, textual data is standardized through lowercasing and removal of unnecessary symbols.

```python
def clean_text_nlp(text):
    if not isinstance(text, str):
        return ""

    text = text.lower()
    text = re.sub(r"http\S+|www\S+", " url ", text)  # keep URL signal
    text = re.sub(r"[^a-z0-9\s]", " ", text)        # keep numbers
    text = re.sub(r"\s+", " ", text)
    return text.strip()

df["final_text"] = df["clean_body"].apply(clean_text_nlp)

tfidf = TfidfVectorizer(
    ngram_range=(1, 2),
    max_features=30000,        # allow richer vocab
    min_df=5,                  # remove ultra-noise
    max_df=0.9,                # remove boilerplate
    stop_words=None,           # 🔥 DO NOT REMOVE IMPORTANT WORDS
    sublinear_tf=True,         # 🔥 MASSIVE for RF
    smooth_idf=True,
    norm="l2"
)

X_tfidf = tfidf.fit_transform(df["final_text"])


print("TF-IDF matrix shape:", X_tfidf.shape)
print("Number of TF-IDF features:", len(tfidf.get_feature_names_out()))

tfidf_df = pd.DataFrame(
    X_tfidf[:5].toarray(),
    columns=tfidf.get_feature_names_out()
)

print("\nTF-IDF preview (first 5 emails):")
display(tfidf_df.head())

row_index = 0
row = tfidf_df.iloc[row_index]

top_words = row.sort_values(ascending=False).head(15)

print(f"\nTop TF-IDF words for email index {row_index}:")
print(top_words)
```

Fig 4: TF-IDF Vectorization and Text Preprocessing Implementation

TF-IDF vectorization is then applied to convert text into numerical form based on word importance.

```
TF-IDF matrix shape: (500000, 30000)
Number of TF-IDF features: 30000

TF-IDF preview (first 5 emails):
```

	00	00 00	00 01	00 05	00 09	00 10	00 11	00 12	00 20	00 2001
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 30000 columns

```
Top TF-IDF words for email index 0:
enroncredit           0.200371
enroncredit com       0.163840
credit derivatives    0.159882
paul radous           0.156625
radous                0.154040
rod                   0.152572
of jeff               0.151566
credit support        0.144762
with jeff             0.141695
support for           0.126668
credit                0.126110
nelson                0.121446
derivatives           0.116560
sara                  0.115245
jeff                  0.106344
Name: 0, dtype: float64
```

Fig 5: TF-IDF Feature Matrix with Vocabulary Representation

Both individual words and n-grams are considered to capture contextual relationships. The resulting feature matrix is high-dimensional and sparse, representing each email as a numerical vector suitable for machine learning models.

Enron Emails and PhishTank Data Relationship Analysis

To improve detection accuracy, a relationship analysis is performed between email data and phishing dataset.

```python
def get_valid_targets(target_series):
    valid_targets = []

    for t in target_series.dropna():
        t = t.strip().lower()

        if t == "other":
            continue

        words = re.findall(r'[a-z]+', t)
        valid_targets.extend(words)

    return list(set(valid_targets))

phishing_targets = get_valid_targets(phish_df['target'])

print("Valid phishing targets found:", phishing_targets)

def match_targets(email_text, target_list):
    if pd.isna(email_text):
        return []

    email_text = email_text.lower()
    email_words = set(re.findall(r'[a-z]+', email_text))

    return list(email_words.intersection(target_list))

df['matched_targets'] = df['final_text'].apply(lambda x: match_targets(x, phishing_targets))
df['target_match_count'] = df['matched_targets'].apply(len)
print(
    "Emails mentioning known phishing brands:",
    (df['target_match_count'] > 0).sum()
)
```

```
Valid phishing targets found: ['poste', 'inc', 'george', 'otomoto', 'group', 'cielo', 'accur
Emails mentioning known phishing brands: 363230
```

Fig 6: Phishing Target Extraction and Matching Implementation

Extracted URLs are matched with known phishing URLs, enabling direct identification of malicious emails. Additionally, phishing targets are analyzed by matching keywords associated with impersonated brands within email content. These features provide strong contextual indicators of phishing activity and enhance classification accuracy.

Severity-Based Risk Classification of Emails

A severity-based classification mechanism is introduced to categorize emails into four levels: Low, Medium, High, and Critical. This classification is based on indicators such as urgency-related keywords, phishing target matches, and presence of multiple suspicious features. Emails without indicators are classified as Low, while those with strong indicators are labeled as Critical. This approach improves interpretability and provides a more detailed understanding of phishing risks compared to binary classification.

Severity Distribution of Emails

The analysis of severity distribution shows that most emails fall into Medium and Low categories, while High and Critical categories are less frequent.

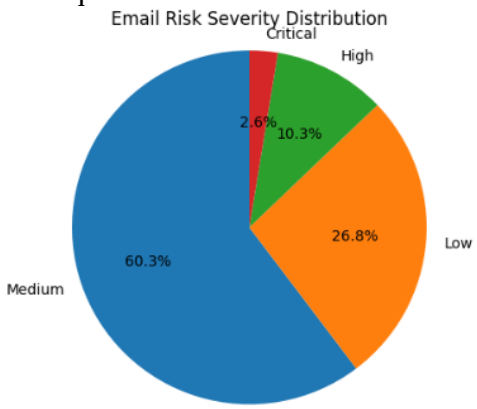

Fig 7: Distribution of Email Risk Severity Levels

This imbalance highlights the need for careful model training and evaluation techniques to ensure accurate classification across all severity levels

E. Model Training & Evaluation

Model Training Process

The dataset is divided into training and testing sets using stratified sampling to maintain class distribution.

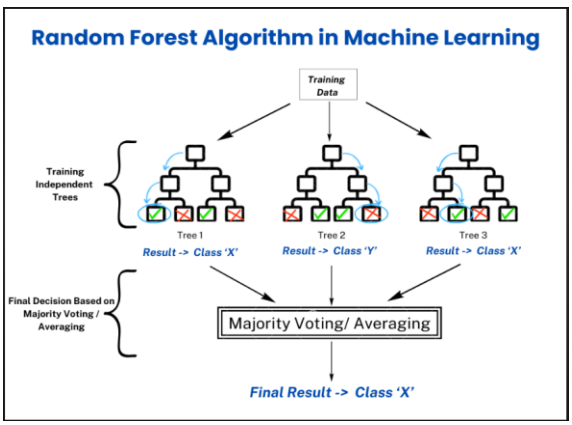

Fig 8: Working of Random Forest Algorithm using Ensemble Learning

A Random Forest classifier is used due to its ability to handle high-dimensional data and provide stable predictions. The model is trained using both TF-IDF features and additional numerical features such as urgency count and target match count. After training, predictions are generated on the test dataset and compared with actual values to evaluate performance.

Model Performance and Evaluation Metrics

The model achieves an overall accuracy of approximately 88%, indicating strong classification performance.

```
from scipy.sparse import hstack

extra_features = df[['urgency_count','target_match_count']].values
X = hstack([X_tfidf, extra_features])

y = df['severity']

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

rf_model = RandomForestClassifier(
    n_estimators=200,
    max_depth=20,
    random_state=42,
    n_jobs=-1
)

rf_model.fit(X_train, y_train)

y_pred = rf_model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n")
print(classification_report(y_test, y_pred))
print("Confusion Matrix:\n")
print(confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.88147

Classification Report:

              precision    recall  f1-score   support

    Critical       0.98      0.61      0.75      2581
        High       0.96      0.39      0.55     10271
         Low       0.99      0.84      0.91     26848
      Medium       0.84      0.99      0.91     60300

    accuracy                           0.88    100000
   macro avg       0.94      0.71      0.78    100000
weighted avg       0.90      0.88      0.87    100000

Confusion Matrix:

[[ 1579   148     0   854]
 [   30  3957     1  6283]
 [    0     0 22646  4202]
 [    1    17   317 59965]]
```

Fig 9: Random Forest Model Training and Performance Evaluation Output

Detailed evaluation using precision, recall, and F1-score shows that the model performs well in Low and Medium severity categories. However, performance in High and Critical categories is relatively lower, mainly due to class imbalance and the complexity of identifying highly suspicious emails.

Confusion Matrix Analysis

The confusion matrix provides insights into prediction accuracy across different severity levels.

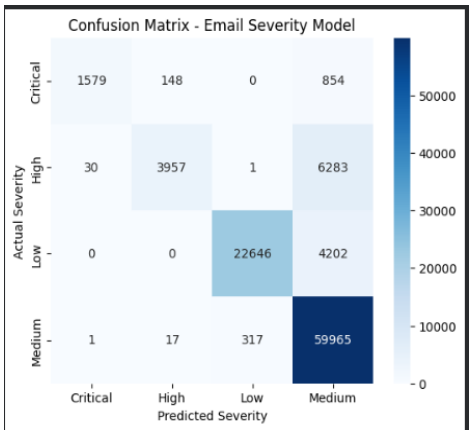

Fig 10: Confusion Matrix for Email Severity Classification Model

It shows that most Low and Medium severity emails are correctly classified, while some misclassifications occur between similar categories such as Medium and High. This indicates overlapping features between severity levels, making classification more challenging.

V. EVALUATION

A. Performance Metrics for Severity Classification

Evaluation metrics such as precision, recall, and F1-score provide a detailed understanding of model performance.

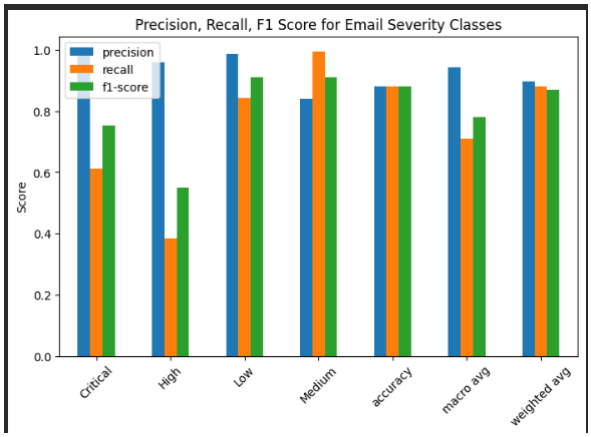

Fig 11: Precision, Recall, and F1-Score for Email Severity Classification

The results indicate high performance in lower severity categories, while High and Critical categories require further improvement. This highlights the importance of addressing class imbalance and improving feature representation.

B. Top Features Influencing Email Classification

Feature importance analysis reveals that urgency count and target match count are the most influential features in classification.

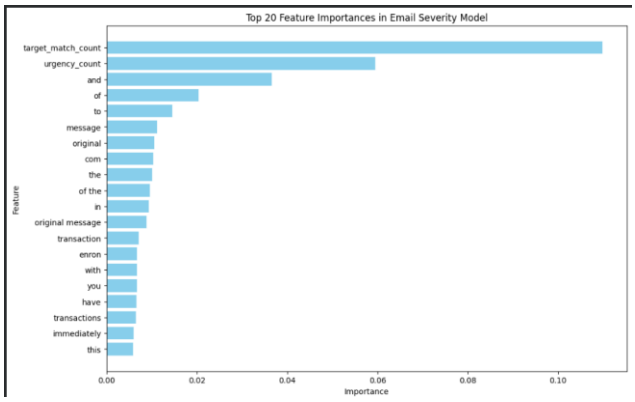

Fig 12: Top Features Influencing Email Severity Classification

Additionally, TF-IDF-based textual features play a significant role in identifying phishing patterns. This demonstrates that both behavioral indicators and textual content are essential for accurate classification.

C. Model Testing Results on Sample Emails

The model is tested on sample email inputs to demonstrate real-world applicability.

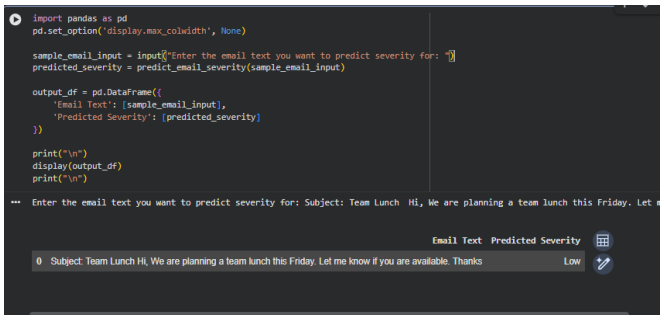

Fig 13: Model Prediction Output for Sample Email Input

A normal email without suspicious indicators is classified as low severity, while emails with moderate risk indicators are classified as medium.

Fig 14: Model Prediction Output for Medium Severity Email

Emails containing urgency language, sensitive requests, and suspicious URLs are correctly classified as high severity.

Fig 15: Model Prediction Output For High Severity Email

These results confirm the effectiveness of the model in distinguishing between different threat levels. Overall, The Proposed System Successfully Integrates Textual Analysis And Contextual Features To Improve Phishing Detection And Severity Classification. It Enhances Interpretability, Reduces Alert Fatigue, And Supports Better Prioritization In Cybersecurity Operations. The Results Demonstrate That Severity-Based Classification Provides Significant Advantages Over Traditional Binary Approaches, Making It Highly Suitable For Real-World SOC Environments.

VI. CONCLUSIONS

This study proposes an advanced Natural Language Processing (NLP)-based framework for phishing email analysis that extends beyond traditional binary classification systems. Instead of simply identifying emails as malicious or legitimate, the proposed model introduces a multi-level severity classification approach, categorizing emails into Low, Medium, High, and Critical levels. This provides a more detailed understanding of cyber threats and enables better prioritization in cybersecurity operations, particularly within Security Operations Centers (SOCs).

The experimental results demonstrate the effectiveness of the proposed approach. The Random Forest classifier achieved an overall accuracy of approximately 88%, indicating strong predictive performance on unseen data. The model performed particularly well in Low and Medium severity classes, achieving high precision and recall with F1-scores up to 0.91. However, performance in High (~0.39 recall) and Critical (~0.61 recall) categories was comparatively lower, mainly due to class imbalance and the complexity of detecting highly suspicious emails.

Further analysis using the confusion matrix shows that most predictions are accurate, especially for Medium severity emails, which form the majority of the dataset (around 60.3%), followed by Low (26.8%), High (10.3%), and Critical (2.6%). This imbalance affects the model's ability to distinguish between closely related categories such as Medium and High. Feature importance analysis reveals that target match count and urgency count are the most influential features, confirming that behavioral indicators, along with TF-IDF-based textual features, play a crucial role in identifying phishing patterns.

Additionally, testing on sample emails demonstrates the model's practical applicability. It correctly classifies normal emails as Low severity, moderately suspicious emails as Medium, and highly suspicious emails with urgency and malicious cues as High severity. Overall, this study addresses a key gap in phishing detection by introducing severity-aware classification, improving threat prioritization, reducing alert fatigue, and enhancing response efficiency in real-world cybersecurity environments.

REFERENCES:

[1] W. S. Admass, Y. Y. Munaye and A. A. Diro, "Cyber security: State of the art, challenges and future directions," Cyber Security and Applications, vol. 2, pp. 1-9, 2024.

[2] O. Alnasser, J. A. M. K. Saleem and S. Shrestha, "Signature and anomaly based intrusion detection system for secure IoTs and V2G communication," Alexandria Engineering Journal, vol. 125, pp. 424-440, 2025.

[3] Supriyono, A. P. Wibawa, Suyono and F. Kurniawan, "Advancements in natural language processing: Implications, challenges, and future directions," Telematics and Informatics Reports, vol. 16, pp. 1-17, 2024.

[4] A. Al-Subaiey, M. Al-Thani, N. A. Alam and A. Khandakar, "Novel interpretable and robust web-based AI platform for phishing," Computers and Electrical Engineering, Vols. 1-23, p. 120, 2024.

[5] A. Orunsolu, A. Sodiya and A. Akinwale, "A predictive model for phishing detection," Journal of King Saud University –Computer and Information Sciences, vol. 34, pp. 232-247, 2022.

[6] P. Alaeifar, S. Pal, Z. Jadidi, M. Hussain and E. Foo, "urrent approaches and future directions for Cyber Threat Intelligence sharing: A survey," Journal of Information Security and Applications, vol. 83, pp. 1-30, 2024.

[7] A. Naz, H. U. Khan, A. Bukhari, B. Alshemaimri and A. Daud, "Machine and deep learning for personality traits detection: a comprehensive survey and open research challenges," Springer, vol. 58, pp. 1-57, 2025.

[8] K. E. Kampourakis, V. Gkioulos, G. Kavallieratos and J.-C. Lin, "Digital Twin-Enabled Incident Detection and Response: A Systematic Review of Critical Infrastructures Applications," Springer, vol. 24, pp. 1-42, 2025.

[9] R. Tanti, "Study of Phishing Attack and their Prevention Techniques," INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT, pp. 1-44, 2024.
[10] P. Kearneya, M. Abdelsame and X. Schmoora, "Combating Alert Fatigue in the Security Operations Centre," Birmingham City University, School of Computing and Digital Technology,, Birmingham, 2025.
[11] L. C. elik, N. Amirov, E. A. Caner, E. Yurdakul and Fahri Anıl Yerlikaya, "Enhancing Phishing Detection in Financial Systems through NLP," Science Direct, pp. 1-11, 2025.
[12] S. Sallouma, T. Gaber and S. vadera, "Phishing Email Detection Using Natural Language Processing echniques: A Literature Survey," Science Direct, vol. 189, pp. 19-28, 2021.
[13] S. Dey, W. Sarma and S. Tiwari, "AI-powered phishing detection: Integrating natural language processing and deep learning for email security," World Journal of Advance Engineering Technology and Sciences, vol. 10, no. 2, pp. 394-415, 2023.
[14] A. Mittal and D. D. Engels, "Phishing Detection Using Natural Language Processing and Machine Learning," SMU Datascience Review, vol. 6, pp. 1-30, 2022.
[15] S. Kavya and D. Sumathi, "Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection," Springer, vol. 58, no. 50, pp. 1-46, 2025.
[16] H. U. Khan, R. A. Khan, H. S. Alwageed and A. O. Almagrabi, "AI-driven cybersecurity framework for software development based on the ANN-ISM paradigm," Scientific Reports, vol. 15, no. 13, pp. 1-45, 2025.
[17] D. Mladenovic, M. Antonijevic, L. Jovanovic and V. Simic, "Sentiment classification for insider threat identification using metaheuristic optimized machine learning classifiers," Scientific Report, vol. 41, pp. 14-25, 2024.
[18] A. H. Salem, S. M. Azzam, O. E. Emam and A. A. Abohany, "Advancing cybersecurity: a comprehensive review of AI-driven detection techniques," Journal of Big Data, vol. 11, no. 105, pp. 1-38, 2024.
[19] M. A. Ferrag, F. Alwahedi, A. Battah and A. Mechri, "Generative AI and Large Language Models for Cyber Security: All Insights You Need," arXiv, 2024.
[20] S. Silvestri, S. Islam, D. Amelin, G. Weiler and S. Papastergiou, "Cyber threat assessment and management for securing healthcare ecosystems using natural language processing," Journal of Information Security, vol. 23, no. 39, pp. 31-50, 2024.
[21] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," Journal of King Saud University - Computer and Information Sciences and Information Sciences, vol. 35, no. 2, pp. 590-611, 2023.
[22] W. LI, S. MANICKAM, W. LENG and P. NANDA, "A State-of-the-Art Review on Phishing Website Detection Techniques," IEEEAccess, pp. 1-37, 2024.
[23] S. Silvestri, S. Islam, S. Papastergiou and C. Tzagkarakis, "A Machine Learning Approach for the NLP-Based Analysis of Cyber Threats and Vulnerabilities of the Healthcare Ecosystem," Sensors, vol. 23, pp. 1-26, 2023.
[24] M. Darwich and M. Bayoumi, "An evaluation of the effectiveness of machine learning prediction models in assessing breast cancer risk," Informatics in Medicine Unlocked, vol. 49, pp. 101-105, 2024.
[25] M. A. Ferrag, F. Alwahedi, A. Battah and B. Cherif, "Generative AI in cybersecurity: A comprehensive review of LLM applications and vulnerabilities," Internet of Things and Cyber-Physical Systems, vol. 5, pp. 1-46, 2025.
[26] A. K. Jha, “Sensing and Supervising through IoT,” International Journal of Computer Applications, vol. 152, no. 9, pp. 7–9, 2016. ISSN: 0975-8887. DOI: 10.5120/ijca2016911723.

[27] A. K. Jha, M. P. Patel, and T. D. Pawar, "Fog offloading: Review, research opportunity and challenges," in Proc. 2019 Int. Conf. Smart Syst. Invent. Technol. (ICSSIT), 2019, pp. 1224–1227. DOI: 10.1109/ICSSIT46314.2019.8987905.

[28] A. K. Jha, M. P. Patel, and T. D. Pawar, "A proposed model of computation offloading in fog environment," Sambodhi (UGC Care Journal), vol. 43, no. 03(IV), pp. 1–6, Nov.–Dec. 2020. ISSN: 2249-6661.

[29] A. K. Jha and T. Pawar, "Computation Offloading for Smart Healthcare Applications," in IoT Applications for Healthcare Systems. Cham: Springer, 2022, pp. 121–136. DOI: 10.1007/978-3-030-91096-9_7.

[30] A. K. Jha, M. P. Patel, and T. D. Pawar, "Computation offloading using K-nearest neighbour time critical optimisation algorithm in fog computing," International Journal of Wireless and Mobile Computing, vol. 23, no. 3–4, pp. 281–292, 2022. ISSN: 1741-1084 (Print), 1741-1092 (Online). DOI: 10.1504/IJWMC.2022.127593.

[31] A. K. Jha, M. P. Patel, and T. D. Pawar, "Extended hybrid cluster algorithm for computation offloading in fog computing," International Journal on Technical and Physical Problems of Engineering (IJTPE), issue 51, vol. 14, no. 2, pp. 176–182, Jun. 2022. [Online]. Available: https://www.iotpe.com/IJTPE/IJTPE-2022/IJTPE-Issue51-Vol14-No2-Jun2022/21-IJTPE-Issue51-Vol14-No2-Jun2022-pp176-182.pdf

[32] M. Patel, A. Mehta, A. K. Jha, A. Patel, and A. Nayak, "A deep reinforcement prediction model for live VM migration in fog," International Journal on Technical and Physical Problems of Engineering (IJTPE), issue 58, vol. 16, no. 1, pp. 277–283, Mar. 2024. [Online]. Available: https://www.iotpe.com/IJTPE/IJTPE-2024/IJTPE-Issue58-Vol16-No1-Mar2024/40-IJTPE-Issue58-Vol16-No1-Mar2024-pp277-283.pdf

[33] V. Soni and A. Jha, "IoT Botnet Attacks Detection Using Deep Learning Approaches: A Review," IET Conference Proceedings, vol. 2025, no. 7, pp. 253–260, 2025. (IET Conf. Proc. series are typically indexed in Scopus, Ei Compendex, and IET Inspec—though explicit confirmation for this exact issue is not available) IET Events.

[34] R. Shankar, I. Kumar, M. Kashyap, A. K. Jha, and B. P. Chaudhary, "A Review on NOMA scheme for emerging 6G wireless networks: State of the Art, Key Schemes, Future scope and Security Issues," Radioelectronics and Communications Systems, vol. 68, no. 5, pp. 271–284, 2025. DOI: 10.3103/S0735272725010017.

[35] M. S. Shaikh, A. K. Jha, Y. Singh, N. K. Jain, A. Seal, and A. Pandwal, "Beyond Traditional Prenatal Monitoring: An Intelligent IoT-Based Pre-eclampsia Detection," in EPJ Web Conf., vol. 328, p. 01062, 2025. DOI: 10.1051/epjconf/202532801062.

[36] M. S. Shaikh, A. K. Jha, B. R. Soni, R. N. K. Patel, and D. P. M., "Flying Edge Intelligence: UAV-Driven Edge Computing for Autonomous Precision Farming," in Proc. 2025 Int. Conf. Emerging Technol. Eng. Appl. (ICETEA), 2025, pp. 1–6. DOI: 10.1109/ICETEA64585.2025.11099749.