Risk Informed Network Threat Response and Analysis

[1*]M. Arul sankar, [2]A. Ashwini, [3]V. Atshaya, [4]D. Kavya, [5]G. Mathivarshni

[1,2,3,4,5]Department of Information Technology, Mahendra Engineering College (Autonomous), Namakkal, Tamilnadu, India.

Abstract

With the increasing use of digital platforms for communication, banking, shopping, and government services, the number of online threats has also grown rapidly. Users are frequently exposed to scam messages, phishing links, and misleading content that appear legitimate at first glance. These threats often use urgency, fear, or attractive offers to manipulate users into revealing sensitive information. The major challenge is that most users are not able to clearly distinguish between genuine and malicious content, which leads to a high number of cyber fraud cases. This project presents RINTRA (Risk Informed Network Threat Response and Analysis), a web-based system. The system allows users to input suspicious messages or URLs and evaluates them using a combination of techniques, including keyword-based analysis, statistical pattern detection, dark pattern recognition, and external API-based URL scanning. These methods work together to identify indicators of phishing, financial fraud, identity theft, and manipulative design practices. To provide a more accurate assessment, the system calculates a composite risk score on a scale from 0 to 100 by combining multiple factors such as message content, URL safety, and behavioral patterns. Instead of only presenting technical results, RINTRA also uses an AI-based model to generate a clear and understandable explanation of the detected threat. It explains what is happening, why the content is considered risky, and what actions the user should take, such as avoiding the link or reporting the message. The system is designed with a focus on usability and accessibility, making it suitable for both technical and non-technical users. It bridges the gap between complex cybersecurity tools and everyday user needs by simplifying threat detection and decision-making. While the system performs well in identifying common types of scams and phishing attempts, its effectiveness depends on the quality of input data and external API responses.

Keywords — Cybersecurity, Scam Detection, Phishing Analysis, Threat Detection, Risk Scoring, Dark Pattern Detection, URL Scanning, Online Fraud Prevention.

1. INTRODUCTION

In recent years, digital communication has become a normal part of everyday life. People use their phones and computers for banking, shopping, paying bills, and even accessing government services. While this shift has made life easier, it has also opened the door to a growing number of online threats. Many users receive messages that look genuine but are actually designed to trick them into sharing personal or financial information. A common example is receiving a message that claims a bank account has been blocked or that a payment is pending, followed by a link to “verify” details. These messages often create a sense of urgency so that users act quickly without thinking. The problem is not just the existence of these scams, but how convincing they have become. Even educated users sometimes fail to identify them correctly.

There are tools available to detect such threats, but most of them are either too technical or not user-friendly. Some tools only focus on URL checking, while others require manual investigation. In many cases, users are left with a result like “malicious” or “safe” without any proper explanation. This creates confusion instead of helping the user make a clear decision.

This project, RINTRA (Risk Informed Network Threat Response and Analysis), is developed to address this gap. The idea behind RINTRA is simple, instead of expecting users to understand cybersecurity concepts, the system should analyze the content and explain the risk in a way that anyone can understand. The system takes a

suspicious message or link as input and evaluates it using multiple approaches, including keyword analysis, pattern recognition, and real-time URL (Uniform Resource Locator) scanning. It then generates a risk score and provides a clear explanation of what is happening and why the content may be dangerous. The main goal of this project is to make threat detection more practical and accessible. Instead of building a complex system only for experts, RINTRA is designed to help everyday users make safer decisions when dealing with online content.

2. Related Works

In the area of cybersecurity, a lot of work has already been done on detecting phishing messages, malicious links, and suspicious online behavior. Most of these systems focus on one specific part of the problem, such as URL filtering, email scam detection, or malware scanning. Some of them perform well in a controlled setting, but they often fail when the input is written in a more realistic way, especially when scammers use simple language, urgency, or emotional pressure to trick users. This is one of the main reasons why threat detection still remains a practical challenge even though many tools already exist.

Several earlier studies and tools mainly rely on keyword matching or rule-based filtering to identify suspicious content. These methods are easy to build and fast to run, but they are limited because attackers keep changing the wording of scam messages. A message may not contain obvious scam words and still be dangerous. In many cases, fraudulent messages are written in a way that looks normal at first glance, which means keyword-based systems alone are not enough. They can catch some basic threats, but they usually miss more subtle manipulation techniques.

Another important area of work is URL reputation checking. Tools like online link scanners and multi-engine security platforms inspect a URL and compare it against known malicious databases. These systems are useful because they can identify dangerous websites based on their behavior, domain structure, or scan results from multiple engines.

However, they are not always enough on their own. A URL may look suspicious even before it is reported anywhere, and some fake websites are newly created, which means reputation-based tools may not detect them immediately. Because of this, many researchers now combine URL scanning with local heuristic analysis to improve detection.

Some studies have also focused on identifying dark patterns in digital interfaces and scam messages. Dark patterns are manipulative techniques used to pressure users into making quick decisions, such as fake urgency, hidden charges, false offers, or forced consent.

These patterns are especially common in scam websites, fake shopping pages, and misleading subscription prompts. Research in this area has shown that users often respond to pressure-based design without fully reading the content. This means that detecting only the technical threat is not enough; the system also needs to recognize the manipulation style used in the message or page.

3. Methodology

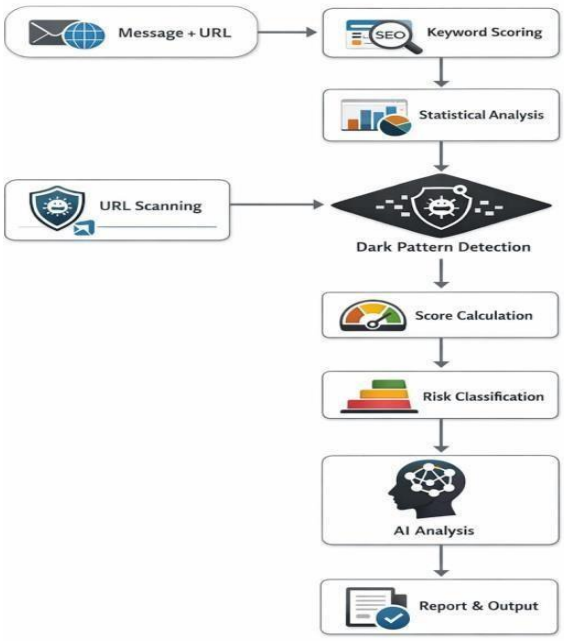

Fig 1: Workflow for URL-Based Detection

A. Input Processing and Preparation

The methodology begins with collecting input from the user, which can be a suspicious message, a URL, or a combination of both. In real-world scenarios, such inputs are often unstructured and inconsistent, as scam messages are intentionally designed to confuse or mislead users. They may include random capitalization, excessive punctuation, shortened links, or mixed formatting. Because of this, the system first performs a preprocessing step to make the input suitable for analysis. This involves normalizing the text, identifying key components such as embedded links, and preparing the data in a consistent format. This stage may seem simple, but it plays a crucial role in ensuring that the later stages of analysis work accurately and do not miss important patterns due to formatting variations.

B. Message and Pattern Analysis

After preprocessing, the system performs detailed analysis of the message content using a combination of heuristic and statistical techniques. The primary method used here is heuristic keyword scoring, where a predefined set of threat-related keywords is used to evaluate the message. Each keyword is assigned a weight based on its severity, and the total message score is calculated as:

Message Score = ∑ (Keyword Weight)

+ ∑ (Pattern Contributions)

In addition to keyword detection, the system applies statistical text analysis to capture indirect indicators such as phone numbers, excessive capitalization, repeated punctuation, and currency symbols. Each of these contributes a small incremental score, strengthening detection even when obvious keywords are not present. The system also integrates regex-based dark pattern detection, which identifies manipulative techniques such as false urgency, forced consent, and misleading offers. These patterns are detected using predefined expressions, and each match increases the overall pattern score. By combining these approaches, the system produces a message-level risk score that reflects both direct and indirect indicators of suspicious behavior.

C. URL Analysis

When a URL is present, the system performs a separate analysis to evaluate its safety. This involves both external and internal methods. The external method uses a multi-engine scanning service, which checks the URL across multiple security systems and returns classifications such as malicious or suspicious. The internal analysis applies heuristic checks on the structure of the URL, such as identifying suspicious domains, excessive subdomains, and unusual top-level domains.

A key algorithm used in this stage is Shannon Entropy, which measures the randomness of the domain name. The entropy is calculated as:

$$\mathbf{H} = -\sum \mathbf{p(x)} \mathbf{log_2\, p(x)}$$

D. Where ($\boldsymbol{x}$)represents the probability of each character in the domain. Higher entropy values indicate randomness, which is commonly found in malicious or automatically generated domains. If the entropy exceeds a predefined threshold, the URL is considered suspicious. The system is designed using a client-server architecture where the frontend and backend work together total engines to perform real-time threat analysis. The frontend is developed using React.js and is responsible for user interaction and initial analysis, while the backend is built using FastAPI (Application Programming Interface) and handles external API communication and AI (Artificial Intelligence). This ensures that malicious detections have a stronger impact than suspicious ones.Risk Scoring Mechanism

After obtaining individual scores from message analysis, URL evaluation, and pattern detection, the system combines them into a single composite score. This is done using a weighted scoring algorithm that ensures balanced decision-making.

When both message and URL are available, the composite score is calculated as:

Composite Score = ($\mathbf{0.40}$ × Message Score) + ($\mathbf{0.40}$

× URL Score) + ($\mathbf{0.20}$ × Pattern Score)

If no URL is present, the system adjusts the weights: Composite Score = ($\mathbf{0.70}$ × Message Score) + ($\mathbf{0.30}$

× Pattern Score)

This adaptive weighting ensures that the system uses available information effectively without over-relying on any single component. The final score is normalized between 0 and 100, where higher values indicate higher risk.

E. Result Interpretation and User Guidance

The final stage of the methodology focuses on how the results are presented to the user. Instead of displaying only technical details or raw scores, the system provides a clear and understandable explanation of the analysis. This explanation describes the nature of the detected threat, the possible intent behind the message or link, and the reasons why it has been classified as risky.

In addition to explaining the risk, the system also offers practical guidance on what the user should do next. This may include avoiding interaction with the content, not clicking on suspicious links, or reporting the message through appropriate channels. This step is important because detection alone does not solve the problem; users need clear direction to make safe decisions. By combining analysis with explanation and guidance, the system ensures that it is not only technically effective but also practically useful for everyday users.

The system follows a sequential data flow. When a user inputs a suspicious message or URL, the frontend first performs lightweight analysis such as keyword scoring, statistical checks, and dark pattern detection. If a URL is present, it is sent to the backend, which forwards it to the VirusTotal API for multi-engine scanning. The results from all components are then combined to compute a final risk score. After scoring, the data is passed to the AI module for interpretation, and the final output is displayed to the user.

F. Processing and API Handling

The system is developed using FastAPI and serves as the central processing layer where all major operations are coordinated. It acts as the bridge between the user interface and the external services required for threat analysis. One of the most important roles of the backend is to manage communication with external APIs such as the URL scanning service and the AI model. By handling all external requests internally, the system maintains better control over security and prevents misuse or unauthorized access. The backend also plays a key role in processing and organizing the data received from different components. When a URL is submitted, the backend forwards it to the external scanning service and waits for the analysis to be completed.

Once the results are received, they are processed and converted into a structured format that can be easily used by the rest of the system. Similarly, after the frontend completes initial analysis and scoring, the backend takes this data and sends it to the AI module for further interpretation. Another important responsibility of the backend is to coordinate multiple operations without interrupting the user experience. It handles asynchronous API calls, manages response timing, and ensures that all parts of the system work together smoothly. This is especially important when dealing with external services that may take time to return results. The backend ensures that the system remains responsive while waiting for these operations to complete.

G. URL Scanning Module

For URL analysis, the system integrates with the VirusTotal API to evaluate the safety of links provided by the user. When a URL is submitted, the backend forwards it to the VirusTotal service, which performs a comprehensive scan using multiple security engines. Each engine independently analyzes the URL and provides its own verdict, such as whether the link is malicious, suspicious, or safe. The backend then collects these individual responses and summarizes them into a structured result that reflects how many engines flagged the URL under each category. This approach allows the system to rely on a wide range of security sources instead of depending on a single detection method. Since different engines use different detection techniques, combining their results provides a more reliable and balanced assessment. The backend processes this data carefully and prepares it in a form that can be used for further scoring and analysis within the system.

Another important advantage of this integration is

that it provides real-time threat intelligence. If a URL has already been identified as harmful by any of the scanning engines, the system can immediately reflect that in its analysis. This makes the detection process faster and more accurate, especially for known phishing sites or previously reported malicious links. At the same time, the system is designed to handle cases where a URL may not yet be widely recognized as harmful. Even if only a few engines flag the link as suspicious, the system still considers this information during risk calculation. By combining these results with other analysis methods, the system improves its ability to detect potential threats more effectively.

H. AI Integration Module

The AI module in RINTRA is integrated into the backend using the Groq LLaMA 3.3 70B model and plays a key role in making the system more user-friendly. Unlike traditional systems where AI is used directly for detecting threats, in RINTRA the detection is already handled by earlier stages such as message analysis, URL scanning, and scoring algorithms. The purpose of the AI here is not to identify threats from scratch, but to interpret the results generated by these components and present them in a meaningful way. Once the system completes the analysis, all the relevant data is collected in a structured format. This includes the original message, URL scan results, individual scores from different modules, and the final composite risk score. This structured input is then sent to the AI model through the backend. Because the AI receives all the necessary context together, it is able to understand the overall situation instead of relying on isolated values.

Based on this input, the AI generates a detailed explanation that helps the user understand what is happening. It provides a clear summary of the type of threat, explains why the content is considered risky, and describes the reasoning behind the assigned severity level. In addition to this, it also suggests practical actions that the user should take, such as avoiding interaction with the content or reporting it through appropriate channels. This approach significantly improves the usability of the system. Instead of presenting only numerical scores or technical outputs, RINTRA converts the analysis into simple and understandable language. This makes the system accessible even to users who do not have a technical background, allowing them to make informed decisions without confusion. Overall, the AI module acts as a bridge between complex analysis and user understanding, enhancing both clarity and effectiveness of the system.

I. Data Flow and Processing Pipeline

The system operates using a structured and sequential data flow, where each stage contributes to the overall analysis in a coordinated manner. The process begins at thefrontend, where the user provides input in the form of a message, a URL, or both. As soon as the input is received, the frontend performs initial analysis, which includes basic text processing and detection of patterns such as keywords or suspicious formatting. This step helps in preparing the data and extracting useful information before involving the backend. If the input contains a URL, it is then sent to the backend through an API request for further evaluation. The backend takes this URL and forwards it to an external scanning service, which performs a detailed analysis using multiple security engines. Once the scanning process is complete, the backend receives the results and processes them into a structured format. These results are then sent back to the frontend so they can be combined with the message-level analysis.

After receiving all the necessary data, the frontend computes the composite risk score by combining the message score, URL score, and pattern-based indicators. This step is important because it brings together different aspects of the analysis into a single value that represents the overall risk level. Once the scoring is completed, the system proceeds to the interpretation stage. The complete set of data, including the original input, analysis results, and computed scores, is sent back to the backend for AI-based processing. The backend then passes this structured information to the AI model, which generates a detailed explanation of the findings.

J. Output Generation and Reporting

The final output of the RINTRA system is designed to present the analysis results in a clear and meaningful way so that users can easily

understand the level of risk associated with the given input. Instead of displaying only raw numerical values, the system combines multiple elements to provide a complete view of the analysis. This includes the composite risk score, which represents the overall level of threat, along with specific indicators that were detected during the analysis process, such as suspicious keywords, patterns, or risky URL characteristics. In addition to these technical results, the system also provides an AI-generated explanation that describes the situation in simple terms. This explanation helps the user understand why a particular message or link has been classified as risky and what factors contributed to that decision. By presenting both the score and the reasoning together, the system ensures that the user is not left guessing or confused about the result.

The output is displayed in a structured format that makes it easy to interpret at a glance. Users can quickly identify whether the content is safe, suspicious, or highly risky, and take appropriate action based on the provided guidance. This is especially useful for non-technical users who may not be familiar with cybersecurity concepts. In addition to real-time analysis, the system also includes a report generation feature. Users can download a detailed report of the analysis, which contains all relevant information such as input data, detected indicators, risk scores, and explanations. This feature is useful for record-keeping, sharing results with others, or using the report as supporting evidence when reporting suspicious activity.

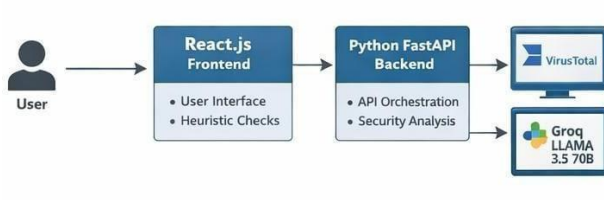

Fig 2: Web Based Dark Pattern Detection

4. RESULT AND DISCUSSION

The system was evaluated using a carefully designed set of test inputs that simulate a wide range of real-world communication scenarios. These inputs included phishing messages, financial fraud alerts, impersonation-based messages, normal conversational text, and mixed inputs containing both message content and URLs. The objective of this evaluation was not only to verify whether the system can detect malicious content, but also to analyze how consistently it performs across different input categories. Particular focus was given to understanding system behavior in situations where the intent of the message is clear, as well as cases where the input is ambiguous or partially suspicious.

The test dataset was structured in such a way that it included both straightforward and complex scenarios. Straightforward cases involved messages with obvious indicators such as urgency, financial incentives, or direct requests for sensitive information. Complex cases included messages that were intentionally written in a subtle manner, avoiding common scam keywords while still attempting to influence user behavior. By testing the system against this range of inputs, it was possible to observe how well the different components of the system contribute to the final decision-making process. During evaluation, the system showed strong performance in detecting clearly malicious messages. Messages that contained urgency-based phrases, such as requests to act immediately or warnings about account suspension, were consistently assigned high message scores. Similarly, messages involving financial elements, such as refunds, rewards, or payment requests, triggered higher scores due to the presence of high-weight keywords. The statistical analysis component further strengthened detection by identifying additional signals such as excessive capitalization, presence of phone numbers, and unusual punctuation patterns. These combined effects ensured that messages with strong malicious intent were accurately classified.

The integration of URL analysis significantly improved the system's ability to detect threats that are not immediately visible from the message content alone. When a URL was present in the input, it was analyzed using external multi- engine scanning, which provided a detailed assessment of the link's safety. URLs that were flagged as malicious by multiple engines resulted in a

substantial increase in the overall risk score. Even in cases where the URL was marked as suspicious by only a few engines, the system incorporated this information into the final score, reflecting a moderate level of risk. This behavior demonstrates that the system effectively balances the influence of different detection signals rather than relying solely on a binary classification.

An important observation from the results is the effectiveness of the system's multi-layered analysis approach. Instead of depending on a single detection technique, the system combines heuristic keyword scoring, statistical pattern recognition, dark pattern detection, and URL intelligence. This layered approach allows the system to capture both explicit and implicit indicators of risk. For example, a message that does not contain strong keywords but includes a suspicious link can still be classified as high risk due to the contribution of URL analysis. Similarly, messages that use manipulative language without obvious malicious content can still be flagged through pattern detection mechanisms.

The dark pattern detection component proved particularly useful in identifying subtle forms of manipulation. Messages that used techniques such as false urgency, limited-time offers, or forced actions were observed to receive additional score contributions. These patterns are commonly used in modern scams, where attackers rely on psychological pressure rather than direct threats. By detecting these behavioral cues, the system extends its detection capability beyond traditional rule-based methods and captures a broader range of suspicious activities.

At the same time, the system demonstrated reliable performance when processing safe and legitimate inputs. Messages that did not contain suspicious keywords, unusual patterns, or risky URLs were consistently assigned low risk scores. This indicates that the system maintains a balance between sensitivity and accuracy, avoiding unnecessary false alarms. This aspect is particularly important in practical usage, as users are more likely to trust and adopt a system that does not generate excessive false positives. The observed results suggest that the system is capable of distinguishing between normal communication and potentially harmful content with reasonable accuracy.

However, the evaluation also revealed certain limitations that need to be considered. Messages that are carefully designed to avoid obvious indicators may not be strongly detected by the system and are often classified under medium risk. While this cautious approach reduces the risk of false negatives, it introduces some level of uncertainty in borderline cases. In such situations, the system relies on the user's judgment to make the final decision. This highlights the need for further improvement in handling subtle and evolving threat patterns.

Another limitation is related to the dependency on external services for URL analysis. Since the system relies on external APIs for multi-engine scanning, the response time and availability of these services can affect overall performance. In cases where the external service is slow or temporarily unavailable, the system falls back to local heuristic checks. Although this ensures continuity of operation, the accuracy of local checks may not match that of real-time external scanning. This trade-off between availability and accuracy is an important consideration in the system design.

The scoring mechanism used in the system plays a crucial role in combining different sources of information into a single meaningful value. The weighted aggregation approach ensures that both message content and URL behavior are given appropriate importance. By assigning different weights to different components, the system avoids over-dependence on any single factor. This results in a more balanced and realistic assessment of risk. The normalization of scores within a fixed range further improves interpretability, allowing users to easily understand the severity of the threat. The AI-based interpretation module adds significant value to the overall system by improving user interaction and understanding. Instead of presenting only numerical scores, the system provides a clear explanation of the detected threat, including the reasoning behind the classification and suggested actions. This is particularly useful for non-technical users, who may find it difficult to interpret raw analysis results. The AI module

effectively translates complex outputs into simple language, making the system more accessible and practical for everyday use.

5. CONCLUSION

This project was developed to address the increasing issue of online scams, phishing messages, and misleading digital content by providing a practical and user-friendly threat analysis system. The system combines multiple techniques such as keyword-based message analysis, statistical pattern detection, URL scanning through external services, and AI- based interpretation to evaluate suspicious inputs in real time. One of the key strengths of the system is its multi-layered approach. Instead of relying on a single method, it integrates different forms of analysis to produce a more reliable and balanced risk assessment. The use of a weighted scoring mechanism ensures that both message content and URL behavior are considered, resulting in a comprehensive evaluation. In addition to detection, the inclusion of an AI- based explanation module improves usability by converting technical results into clear and understandable insights. The results indicate that the system performs effectively in identifying common types of cyber threats, including phishing attempts, financial scams, and manipulative patterns. At the same time, it maintains a balance by correctly identifying safe inputs without generating excessive false alerts. However, certain limitations were observed, particularly in handling highly subtle or carefully crafted messages and the dependency on external API services for URL analysis. The project demonstrates how combining rule- based techniques, real-time threat intelligence, and AI-based interpretation can create a practical solution for everyday users. It not only detects potential threats but also helps users understand and respond to them, contributing to improved awareness and safer interaction in digital environments.

REFERENCES

[1] A. Karim, M. Shahroz, K. Mustofa, and S. B. Belhaouari, "Phishing detection system through hybrid machine learning based on URL," *IEEE Access*, 2023.

[2] D. Patel and D. Chudasama, "Detection of phishing website using URL," *Journal of Network Security*, 2025.

[3] S. Rao and A. Verma, "Deep learning-based phishing detection using natural language processing," *IEEE Access*, vol. 12, pp. 34567–34580, 2024.

[4] "AI-generated phishing URL detection using reinforcement learning," *IJETT*, 2025.

[5] K. Barik, S. Misra, and R. Mohan, "Web-based phishing URL detection using deep learning optimization techniques," *Int. Journal of Data Science and Analytics*, 2025.

[6] Q. E. Haq et al., "Detecting phishing URLs based on a deep learning approach," *Applied Sciences*, 2024.

[7] R. Vijayakumar, K. P. Soman, and P. Poornachandran, "Deep learning approach for detecting phishing attacks," *IEEE Access*, vol. 11, pp. 70901–70915, 2023.