Cyber Deception And Detection: A Multi-Layer Security Analysis

LAXMI (22BCS059), ROHIT (22BCS100), SAAHIL MISHRA (22BCS105), SHRIYA UDUPA K (22BCS121), DR. SUNIL C. K

COMPUTER SCIENCE DEPARTMENT,
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DHARWAD

Abstract

In an increasingly connected globalized world, a new breed of cyber threat has very much been scaled up in terms of complexity and magnitude. Under the title "Cyber Detection and Deception," this project will investigate and demonstrate the cyberattacks against behavior patterns from both synthetic and real datasets. It emphasizes categories such as ransomware, phishing, botnets, fraud, and social engineering while considering the extent to which system-level measures-like CPU and disk use-can indicate underlying malicious activity. We will look at phishing behavior using URL attributes and user interaction measurements such as click duration and typing latency. Another big contribution is the collection of a real-world Instagram dataset over two months, aimed at a thorough analysis of social engineering attacks. The initial focus of our solution is intuitive data visualization for human-centric cyber threat detection and deception awareness so that analysis is actionable and understood by both technical and non-technical stakeholders.

Introduction

This place speaks in the present form of cyberspace, with every human activity now heavily communicating through the networked systems. This development has made cyber security among the most important fields of research and innovation today. Cyber attacks today are not simply caused by some amateur hackers; most of the time, they are launched by highly structured, long-lasting, and adaptive threat actors. Effects of these attacks, including those causing ransomware to crash a healthcare system and phishing links stealing millions via social engineering, are catastrophic, massive, and growing at an accelerating rate. Though firewall and antivirus solutions are sine qua non, they do not sufficiently detect, prevent, or understand the most advanced forms of contemporary cyber threats.

This project, Cyber Detection and Deception, aims to explore cybercrime analysis from a modern perspective based on behavioral data, real-time system metrics, and user-level interaction signals. Its goal is to identify possible threats that emanate from anomalies and to provide better insights into how deception behaves across platforms and threat surfaces. We do not just resort to black-box classification but emphasize visualization and analytical methods that provide a basis to make threat patterns understandable and interpretable to technical and non-technical audience members alike.

The work is organized around a number of main modules, each corresponding to a different type of cyber threat:

- **System Behavior under Attack:** We analyze the changes in CPU utilization, disk I/O, and memory allocation during simulated ransomware or malware attacks using synthetic datasets. The performance degradation and abnormal spikes that would indicate an ongoing exploitation are visualized by comparing the normal and malicious behavior of the system.
- **User Interaction-Based Phishing Detection:** Phishing is a psychological attack that often relies on manipulating users through emails or fake interfaces.his paper studies the measurable behavioral characteristics of users like clicking speed, typing speed, and hesitation latency concerning normal and phishing scenarios.
- **URL Feature Analysis:** URLs are a common vector for phishing attacks. We analyze the datasets of the URLs such as phished URLs and legitimate URLs. Few features extracted include presence of special characters, length of URLs, number of dots in URLs, and presence of HTTPS, in order to visualize some structural dramatical differences that exist in the phishing websites over the benign ones.
- **Network Traffic and Packet Inspection:** Network packet flows and traffic metadata can highlight access attempts, such as botnets or data exfiltration. Network traces are applied for observing bandwidth

usage, frequency of packet exchanges, and source-destination behaviors that follow the attack.

- **Botnet Detection:** Botnets are networks of compromised systems controlled by a command-and-control (C&C) server. Establishing repetitive traffic patterns, periodic behavior, and suspicious clustering of network activity different from normal client-server communication are found using known botnet datasets.

- **Real-World Instagram Dataset for Social Engineering:** A very specific aspect of our research is the construction of a custom Instagram dataset, collected by hand over a two-month period. The dataset contains conversation threads, interactions with fake profiles, and behavioral observations concerning potential phishing/fraud accounts. Such real-world collections provide very subtle insights into occurrences of deception in a natural setting-a social media environment that everyone uses-compared to synthetic or public datasets.

By means of interactive and layered visualizations, we present new angles for analysis between raw data and the insights derived from cybersecurity. The work serves to detect anomalous behavior and indicates a way to simulate what cyber deception may look like in a real-world environment. Here, that becomes exceedingly useful in training and awareness programs, wherein the GUI helps users understand threats without requiring a technical view of the underlying concepts.

Apart from the analysis of deception, it is also recreated—mostly within the Instagram module, wherein fake accounts were used to observe how phishers lure users into trusting them and giving away sensitive information. This applied approach adds weight to the study of social engineering, complementing its theoretical emphasis in academia.

In essence, our project represents the synergy of data science, human psychology, and cybersecurity under one working framework for proactive detection and intelligent deception. These insights are hoped to feed into future systems that not only detect threats but actively engage in adversarial defense, confusing or delaying attackers to allow critical assets to escape intact.

The following sections present a comprehensive literature review aligned with our methodology, followed by the details of our experimental modules, visualizations, and key insights that emerged from each type of threat investigation.

Background

Cyber attacks are now more automated, more distributed, and more intelligent than ever before. Ransomware is targeting critical infrastructure; phishing has become automatic by bots, and attackers exploit social platforms for reconnaissance and deception. Deepfake technologies and synthetic identity fraud have emerged to further complicate detection. Old line cyber security tools were all about perimeter security, while the new generation of attackers now prey on human behavior and interdependencies among systems. We shall put that insight into play by using machine learning in tandem with behavioral analysis and active deception modeling.

Key Contribution

One of the uses of this study is for creating real-life Instagram data sets for social engineering attacks. Data collection takes place over two months using an Instagram profile created and maintained only for this purpose. Followers added ogathered based on the above profile were organically real users but one or two might not be really real. Activities such as passive watching and engagement allowed for behavioral indicators, interaction threads, and metadata collection that could inform of other ways that phishing attempts may manifest over social spaces.

With this insight, we could track:

- Phishing through the DM and bio
- Patterns in engagement (likes, follow, comment)
- Indications of urgency words and suspicious links
- User behaviors through VPNs according to when they interact

Real-world simulation-based interaction is what makes this dataset real. It is a bridge between synthetic-created databases for cybersecurity and the genuine social media patterns that are unavailable most of the time. This approach to data collection fits perfectly into our visual analytics while at the same time adding unrefined real-world depth to the inquiry into social engineering.

Literature Review

From reactive defense to proactive intelligent systems,' they not only detect an attack but also predict and mislead the threat actor concerned. Our project is titled **Cyber Detection and Deception** the study of different categories of cybercrime-ransomware, phishing, botnet behavior, fraud, and social engineering input: Behavioral features visualized across both synthetic and real datasets. With the present literature review, we intend to provide a deep analytical reflection on the research contributions that inspired our system-level design, feature engineering, visual analysis strategies, and deception techniques.

System-Level Analytics: CPU and Disk Usage Behavior Under Attack

Modern malware creates behavioral footprints to the CPU and I/O subsystems. Kolodenker et al. embarked on the introduction of PayBreak, which utilizes API monitoring to recover keys needed for decrypting ransomware while

simultaneously capturing how much resources were consumed by the encryption threads [1]. Ours was inspired to record CPU spikes, disk I/O volatility, and bursty read-write operations during the synthetic ransomware simulation. This visualizes how resource exploitation occurs during file encryption.

Sommer and Paxson stated that anomaly detection may be promising but has serious issues with high false positives and scalability [2]. We have limited the number of behavioral features extracted and validated them through visualization to enhance anomaly interpretation human-centric, addressing this issue. Our focus is therefore on the understanding of behavioral divergence rather than detection on its own.

Phishing Analysis: Feature Extraction from User Behavior and URL Structure

These publications include content-based, URL-based, and behavior-oriented dimensions of phishing detection. Khonji et al. cite URL obfuscation, image-only email, and tricking DNS as main attack tricks [3]. We further carried on with adding click speed per minutes, typing latency, and bounce rates, which shows that commonly phished users either tend to act fast or hesitate.

Phishing detection has been aided by Basnet et al.'s identification of structural email features in conflicting link text and counts of embedded scripts [4]. Our adaptation included mining URL features such as entropy of the domain name, count of subdomains, and counts of redirect chains. Smadi et al. used natural language processing in the identification of linguistic deception in phishing email [5], prompting us to consider metrics for lexical richness and urgent words in Instagram phishing bios, as well.

Hassan et al. combined CNN models with WHOIS and URL metadata for high-accuracy phishing classification [6]. Although deep learning is not the focus of our work, we extracted similar hybrid features for visualization aimed to help human analysts comprehend feature-class correlations.

Botnet Detection: Behavioral Modeling and Synthetic Replication

Synchronized, stealthy behaviors characterize botnets. Gajbhiye and Lilhore especially highlighted time-oriented and protocol-specific features for the classification of botnets [7]. Our visualization reveals periodic connection attempts, anomalous packet inter-arrival times, and non-random port switching-these being the key indicators of command-and-control behavior.

According to Obaido et al., the benchmark IoT malware dataset was created to replicate attacks on low-resource devices [8]. This dataset compiles resource metrics, such as CPU, memory, and disk I/O usage, which prompted the synthetic emulation of infected nodes. In addition to flow maps, we created temporal histograms that captured the botnet-like activities.

In addition, the botnets copy normal traffic. We visualized such covert traces using temporal clustering and entropy measures, inspired by those earlier works in the literature which advanced stealth evasion models.

Fraud Detection in Transactional Data

Delamaire and others based on fraud detection on supervised models exploiting features such as merchant frequency, amount of transactions, and user location [9]. In this instance, synthetic transaction datasets were used to highlight anomalies in user behavior. The focus was on visuals instead of model accuracy: patterns of excessive micro-transactions or sudden geographic divergence were shown.

Bhattacharyya and others applied ensemble methods on the Credit Card dataset and showed the severe imbalance problem [10]. We addressed this during visualization by using SMOTE sampling and minority class interaction plots. These helped assess if visual features could delineate fraud from normal behavior.

Time-series visualizations were also used in revealing inconsistencies like multiple high-value transactions in quick succession, which were flagged as suspicious.

Network Traffic and Packet Inspection

Moore et al.'s methods of traffic classification are statistical approaches based on byte frequency, average payload size, and flow duration [11]. The same has been done in the experiment, where scattergrams of packets' sizes, bursts, and delays are drawn to find out signatures and stealthy scans for DDoS.

According to Shafiq et al. encrypted traffic could still be analyzed with side channel information such as packet size, timing, and handshake behaviors [12]. In fact, in our case, the packet visualization module uses such metadata to find harmful communication behaviors without any breach of encryption-integri.

Moreover, entropy-based metrics for packet payloads and headers have been introduced, which could help in detecting encoded or tunneled content types-pretty much often used in advanced types of malware and data exfiltration bots.

Instagram-Based Phishing and Social Engineering Dataset Creation

The main contribution of our project is that we have manually created a real-world Instagram dataset which took us months to develop. Inspired by Goga et al., who showed that user metadata can be exploited in order to correlate identities across platforms [13], our objective was to investigate patterns of fake Instagram profiles targeting users by means of deceptive bios, suspicious links, and manipulative messages.

Our dataset accounts for vital deception indicators like urgency words and their variants ("click now", "limited time"), link shorteners in bios, rapid follow-unfollow activity and foreign IP posting times. These features were

annotated manually and their prevalence visualized to find behavioral clusters of phishers.

Apart from this passive collection, we also engage suspected phishers using dummy accounts. This method enables us to manipulate dialogue and study when messages are sent and responses are made-an unprecedented level of behavioral insight rarely found in public datasets.

Cyber Deception and Defensive Camouflage

In cybersecurity, deception consists of creating traps that confuse or delay attackers; for example, Rowe has framed deception evaluating its effectiveness in terms of delay or confusion of the attacker and exhaustion of resources [14]. In this project, deception included engaging the attackers and gathering forensic trails that could be visualized and shared later.

Han et al. employed honeypots in industrial settings to assess attacker dwell time and exploration paths [15]. Whereas we did not employ honeypots, we did simulate interaction traps on Instagram by tempting attackers with the prospect of revealing their strategies.

Goodall et al. and Al-Mousa et al. stressed the role of visual analytics in mitigating alert fatigue and human error [16], [17]. Following this design philosophy, our framework allows analysts to explore CPU anomalies, phishing flows, and social engineering attacks through dashboards.

Data Description and Analysis

In this section, we'll take you through an extensive description of synthetic datasets applied in the research, from validation design to advanced data analysis. The datasets are constructed to capture different facets of behaviors associated with ransomware, phishing targeting, and potentially anomalous network traffic. Below are brief descriptions of some of the identified data sources:

1. Synthetic Ransomware Dataset

Unveils a simulation of a performance log over a period of time. The following are the features:

- **CPU_Usage (%):** Reflects CPU utilization.
- **Disk_Read and Disk_Write (MB/s):** Reflect I/O disk activity.
- **Memory_Usage (%):** Monitors memory utilization.
- **Network_Activity (Mbps):** Throughput on the network.
- **Timestamp:** Minute-wise movement in time, starting from March 1, 2025.
- **Ransomware_Attack:** A binary indicator where 1 means a ransomware event.

The dataset consists of a 2000-minute simulated activity, consisting of random bursts of activities similar to those produced during a ransomware attack. During these windows, significant deviations in resource usage metrics are observed that help illustrate behavioral deviations between normal and attack states.

2. Synthetic Phishing Dataset

The dataset imitates the way a user interacts with a website whereby every clock gives a minute of hours that are actually spent or credited to the website from March 1st, 2024. The following are some of the features:

- **Clicks:** Total clicks made during the session.
- **Time_Spent (seconds):** Duration of activity of the user.
- **Mouse_Movements:** Capture the user's mouse action.
- **Typing_Speeds (WPM):** Define the typing effectiveness.
- **Referral_Source:** The source from which a user reached the site—like Direct, Email, etc.
- **Phishing_Attempt:** A binary label that assigns 1 to phishing.

The abnormal clicks made by artificial phish behavior involve more free clicks but less time in navigation, and they exhibit a rather strange, non-direct essence of linking sources. The techniques of visual analysis in the form of scatter plots, KDE, and violin plots show a strong contrast in the behavior of normal human interaction versus that of simulated phishing attempts.

3. Synthetic Network Traffic Dataset

It starts creating the estimation of the working of a network. This synthetic network traffic dataset is heterogeneous, with numerable feature combinations of mixed distributions:

- **avg_packet_size:** Packet sizes assumed to be normally distributed.
- **duration:** Connection duration assumed to be exponentially distributed.
- **num_connections:** Count data generated by means of Poisson distribution.
- **src_ip_entropy:** Measures the randomness for source IP address and is uniformly distributed.
- **dst_port_count:** Random integers signifying destination port counts.
- **protocol_type:** Categorical variable (TCP, UDP, ICMP) encoded for modeling purposes.

It contains 10,000 records of normal and three genetic attacks—DDoS, brute force, and port scan. Each attack type manifests differently: for example, in DDoS attacks, the number of connections increases; in brute force attacks, the destination port count increases; whereas in port scanning, the source IP entropy rises.

Exploratory Data Analysis (EDA)

Exploratory visualizations like scatter plots, heat maps, and box plots can be used to gain a deeper understanding of the datasets. An Exploratory Data Analysis (EDA) essentially does the following:

- **Reveals correlations** among important metrics.
- **Visualizes distributions** while pinpointing outliers.
- **Validates simulated attack patterns** against observed behavioral anomalies.

Such analyses lay the foundation for subsequent machine learning model development and also help in critically evaluating the nature and quality of the synthetic data itself.

Methodology

This section describes the complete methodology followed in this study. The approach is broken down into several phases, forming a strong pipeline that spans from synthetic data generation to advanced model evaluation and visualization of insights. The main components that define our methodology are as follows:

1. Data Simulation and Generation

Synthetic datasets mimicking realistic cyberattack behaviors are central to this project. The simulation process is structured as follows:

a. Ransomware Dataset Generation System performance data is simulated over a 2000-minute period using `NumPy`. The generated dataset includes metrics such as CPU usage, disk read/write I/O, memory usage, and network throughput. Distinct high-usage bursts are randomly injected to emulate the behavioral signatures of ransomware attacks. Each timestamp corresponding to such abnormal activity is annotated with a binary `Ransomware_Attack` label.

b. User Behavior Phishing Simulation User interactions are modeled on a simulated website, incorporating features like number of clicks, session duration, mouse movements, and typing activity. Abnormal patterns—including high click counts and very short session durations—are deliberately injected to resemble phishing attempts. The `Referral_Source` is also manipulated to reflect suspicious origins during phishing events. These events are labeled using a binary `Phishing_Attempt` indicator.

c. Network Traffic Synthesis The synthetic network traffic dataset is carefully crafted to reflect both normal and malicious behaviors. Various statistical distributions are used for simulation: normal distribution for `avg_packet_size`, exponential for `duration`, Poisson for `num_connections`, and uniform for `src_ip_entropy`. Categorical variables such as `protocol_type` (e.g., TCP, UDP, ICMP) are encoded appropriately. Attack types including denial of service, brute force, and port scanning are emulated by altering key parameters like connection count, destination port access, and IP entropy. This comprehensive dataset is intended for use in multi-class classification tasks.

2. Data Preprocessing and Feature Engineering

Preprocessing and feature engineering form the backbone of preparing synthetic data for modeling. The following steps were undertaken:

- **Data Cleaning:** Removal and handling of missing values and outlier data points to ensure data quality and consistency.
- **Transformation:** Continuous attributes were normalized and scaled. Logarithmic transformations were applied where needed to reduce skewness and improve model performance.
- **Encoding Categorical Features:** Features such as `protocol_type` and `Referral_Source` were transformed into numerical values using `LabelEncoder` to make them compatible with machine learning algorithms.
- **Feature Derivation:** Additional metrics were generated, including:
  - *Forward-Backward Packet Ratios* and *Packet Length Variability* for network traffic.
  - *Cumulative User Engagement Trends* and *Phishing Success Rates* for behavioral analysis.
- **Data Splitting:** The dataset was divided into training (80%) and testing (20%) sets while ensuring the preservation of class distribution within each split.

3. Model Selection and Training

Different machine learning algorithms were employed for classification tasks, each chosen based on the characteristics of the dataset and the intended application.

a. Gradient Boosting Classifier For the network traffic classification task, a Gradient Boosting Classifier with 200 estimators was used. This ensemble method excels in capturing intricate patterns in high-dimensional feature spaces. Boosting, which builds models sequentially with the aim of correcting the errors of the previous ones, contributes to the strong predictive capability of this classifier, especially in scenarios involving subtle behavioral differences and class imbalance.

b. Auxiliary Models for Comparative Analysis To establish performance baselines and enable comparative evaluation, Random Forest and Logistic Regression models were also trained—specifically for the ransomware and phishing datasets. These auxiliary models help in benchmarking the performance of the primary boosting model and shed light on the strengths and limitations of different model architectures when applied to distinct cyberattack behaviors.

4. Model Evaluation and Validation

This stage assesses how well the machine learning models perform using a variety of metrics and diagnostic visualizations. The evaluation particularly focuses on understanding classifier behavior in the presence of class imbalances, as is typical with simulated datasets. Key classification metrics such as accuracy, precision, recall, and F1 score are computed for each class to capture the balance between true positives, false positives, and other factors.

- **Confusion Matrix Visualization:** Displayed as a heatmap, the confusion matrix provides insight into class-wise misclassifications, helping in diagnosing specific failure modes of the model.
- **Feature Importance Analysis:** Quantifies and visualizes the contribution of individual features, enhancing the interpretability of model decisions. This is particularly valuable in identifying which behavioral indicators are most influential in cyberattack classification.

5. Exploratory Data Analysis and Visualization

Visual exploratory analyses will be used for diagnostic and additional spreads of findings throughout the project. Chief among these visualization methods are:

- **Scatter and Line Plots:** Traditional plots that visualize relationships between features, for example, `duration` versus `num_connections` in a network traffic flow, or `clicks` versus `time spent` in a user behavior data metric.
- **Density and Violin Plots:** Useful for comparing distributions of some very important parameters conditions under normal and attack situations when KDE maps and violin maps can come in handy.
- **Correlation Heatmaps:** These maps give intuition into inter-feature dependencies and a hint to potential multi-collinearity via the correlation matrix.
- **Bar Charts:** These are used to highlight proportional differences in the categorical distribution, e.g., types of protocols or referrer sources during phishing attempts.

These techniques will validate the distributions of data and attack patterns that are conjectured under simulations, but on the other side, help to surface certain structures that are latent within the data known to the model trainer.

6. Flow Chart of Methodology

Below is a flow chart that illustrates the complete methodology followed throughout this project:

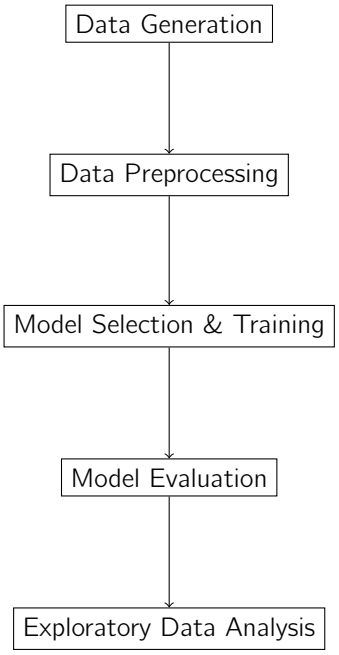

Figure 1: Methodology Flow Chart

The presented approach can comprehensively create, preprocess, train, and evaluate new models for cyberattack detection based on system performance metrics, user behaviors, and network traffic anomalies. Combining synthetic data generation and advanced visualization techniques, this project contributes very much to the understanding of the cyberattack detection pipeline, laying a sound foundation for real-time deployment.

Results and Discussion

Ransomware Attack Simulation: Code and Output Analysis

Ransomware remains a critical cybersecurity threat, often characterized by the encryption of user data followed by ransom demands. To understand its behavior more clearly—and to build better detection strategies—we simulated a ransomware attack in Python using Google Colab. This simulation generates synthetic time-series data for system resource usage during both normal and attack phases. We combined visualizations and statistical methods to analyze the results.

Objective of the Simulation

The primary goals of this simulation include:

- Distinguishing ransomware-induced anomalies from standard system behavior.
- Quantifying ransomware's impact on CPU, memory, disk I/O, and network bandwidth.
- Examining inter-metric correlations during attacks.
- Laying the groundwork for real-time anomaly-based detection.

Simulation Overview

The simulation consists of:

- **Normal Resource Usage:** Modeled using Gaussian distributions to reflect typical activity.
- **Ransomware Behavior:** Sharp resource spikes simulate encryption activity and potential data exfiltration.
- **Visualization Tools:** Time-series plots, attack intervals marked in red, and a correlation heatmap to analyze patterns.

CPU Usage Analysis

Normal Operation: CPU usage averages around 25% with a standard deviation of 8%, showing mild fluctuations.

Attack Phase: CPU surges to 85–100%, usually lasting 10–20 minutes, reflecting intensive encryption tasks.

Inference: CPU usage is a solid anomaly marker due to its clear spike during ransomware activity.

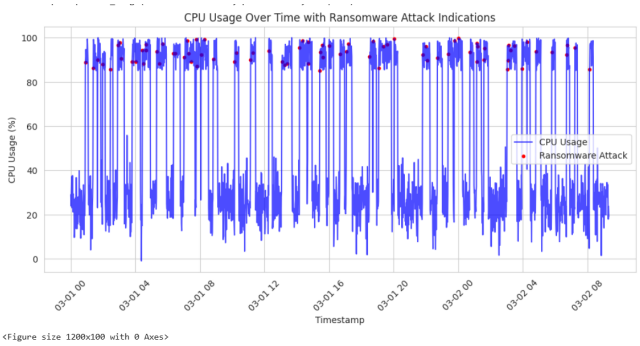

Figure 2: CPU Usage during Normal and Attack Phases

System Resource Analysis

Figure 3: Disk Read and Write during Simulation

Interpretation: Ransomware writes encrypted files, leading to a drastic rise in disk write speeds.

Memory Usage:

- Normal: ∼40%
- Attack: 70–90%

Interpretation: Increased memory use is likely from memory-resident ransomware components and payload execution.

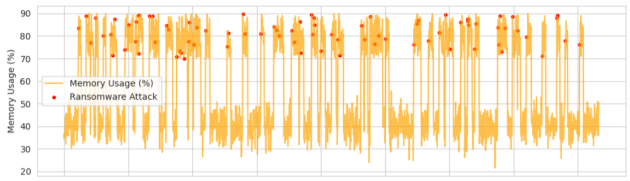

Figure 4: Memory Usage Variation

Network Activity:

- Normal: ∼10 Mbps
- Attack: 50–200 Mbps

Interpretation: Elevated network usage hints at command-and-control (C2) communication or data exfiltration.

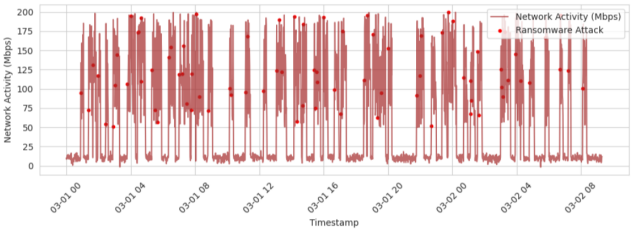

Figure 5: Network Activity Spikes

Metric	Normal	Attack	Relative Increase
Disk Reads	∼15 MB/s	60–120 MB/s	4–8x
Disk Writes	∼8 MB/s	70–160 MB/s	8–20x

Table 1: Quantitative Comparison of Disk I/O Metrics

Correlation Analysis

We used a correlation heatmap to explore relationships between metrics:

Metric Pair	Correlation Coefficient
CPU & Memory	∼0.90
CPU & Disk Write	∼0.88
Disk Write & Network Activity	∼0.91
CPU & Network Activity	∼0.89

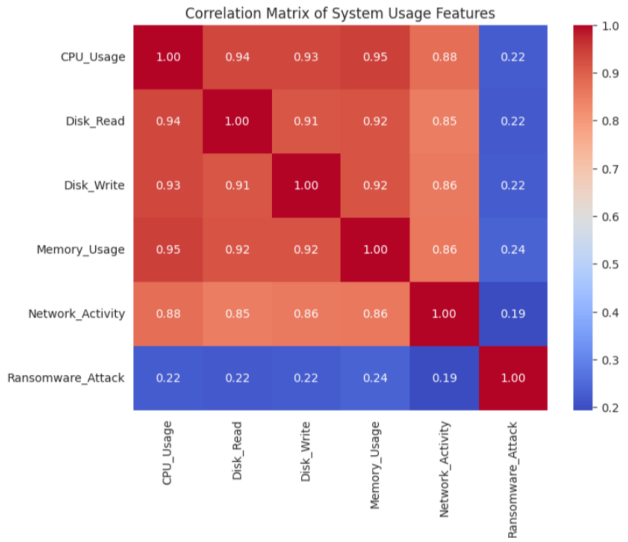

Figure 6: Correlation Heatmap of System Metrics

Insights:

- Strong correlations reflect synchronized resource usage during attacks.
- The clear contrast from normal behavior supports anomaly-based detection.
- Minor negative values may result from plotting artifacts.

Key Findings and Interpretations

Ransomware Signature:

- Brief but significant spikes across all monitored resources.
- Easy to distinguish from regular patterns visually and statistically.

Detection Recommendations:

- Raise alerts for deviations beyond 2–3 standard deviations.
- Combine multiple metrics for accuracy and fewer false alarms.
- Include timing features like duration and spike frequency.

System Impact Summary:

Metric	Normal	Attack	Increase
CPU Usage	∼25%	85–100%	3–4x
Memory Usage	∼40%	70–90%	2x
Disk Writes	∼8 MB/s	70–160 MB/s	8–20x
Network Usage	∼10 Mbps	50–200 Mbps	5–20x

Role of Metric Correlation: Strong cross-metric correlations suggest advanced detection models can be employed:

- Principal Component Analysis (PCA)
- Multivariate Gaussian-based anomaly detection
- LSTM-based temporal deep learning models

This simulation showcases clear behavioral footprints left by ransomware. The drastic and synchronized deviations across system metrics offer solid grounds for designing real-time anomaly-based detectors using multivariate and temporal features.

Future Directions

- Simulate other threats (e.g., spyware, trojans) for comparative insights.
- Shift to real-time system monitoring and stream analytics.

- Train ML models on both synthetic and real-world datasets.

- Examine false positive rates under various system loads.

Phishing Detection via Behavioral Simulation

A Python-based behavioral simulation was conducted to systematically distinguish normal user activity from phishing interactions. The simulation synthesized user behavior across five core interaction metrics: click count, time spent on page, referral source, typing speed, and mouse movement frequency. Visualization libraries such as `Matplotlib` and `Seaborn` were employed to explore these metrics and uncover behavioral fingerprints indicative of phishing activity.

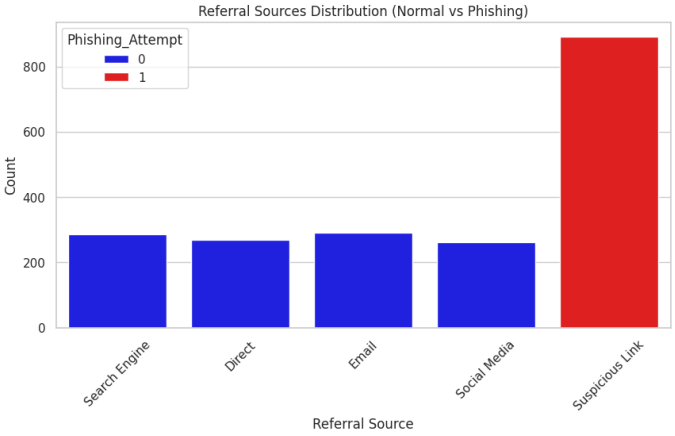

Figure 8: Referral Source Distributions

Key Behavioral Insights

- **Click Frequency vs. Session Duration:** Normal users performed 10 ± 3 clicks over an average of 300 ± 50 seconds, indicating deliberate engagement. In contrast, phishing sessions involved 50–100 clicks within only 30–100 seconds, reflecting urgency or confusion due to deceptive UI prompts. *Insight: A high click rate within a short session window is a strong phishing indicator.*

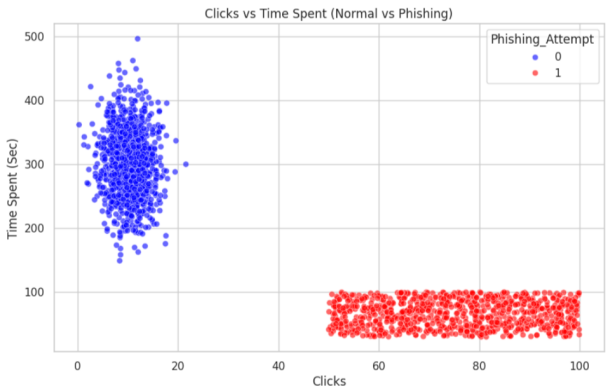

Figure 7: Click Frequency vs. Session Duration

- **Referral Source Analysis:** Legitimate sessions typically originated from trusted sources such as search engines, emails, or social media platforms. Conversely, phishing sessions predominantly began from suspicious or unknown links. *Insight: Referral source is a strong early signal of phishing attempts.*

- **Typing Speed Distribution:** Typing speed in normal sessions followed a normal distribution centered around 40 ± 10 words per minute (WPM). Phishing victims exhibited reduced typing speeds (10–20 WPM), possibly due to hesitation or uncertainty. *Insight: A sudden drop in typing speed during credential entry may signal phishing activity.*

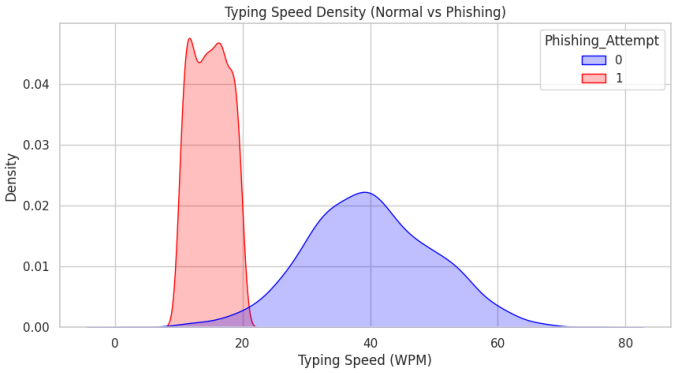

Figure 9: Typing Speed Distribution

- **Mouse Movement Patterns:** Normal sessions featured a high volume of mouse movements (approx. 500 ± 100), indicating active navigation. Phishing interactions were marked by significantly fewer movements (100–200), suggesting user discomfort or scripted actions. *Insight: Reduced or erratic mouse activity may indicate non-genuine interaction.*

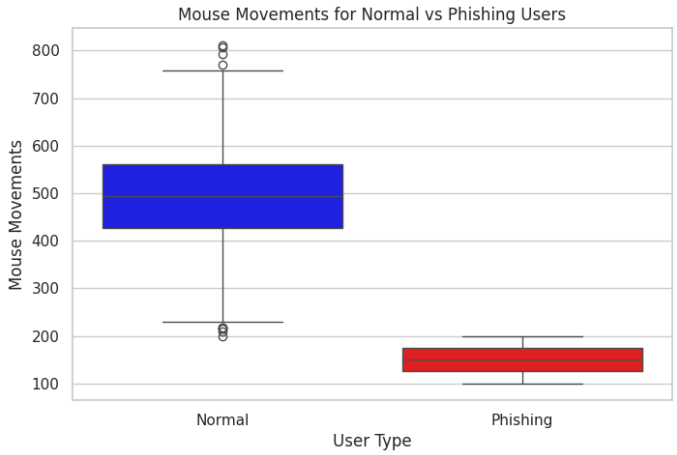

Figure 10: Mouse Movement Patterns

- **Session Duration Trends:** Legitimate user sessions generally lasted between 250–350 seconds, often involving exploration and content interaction. Phishing sessions, on the other hand, were short-lived (usually under 100 seconds) and often terminated abruptly after form submission. *Insight: Sudden session termination immediately after form input is a notable phishing marker.*

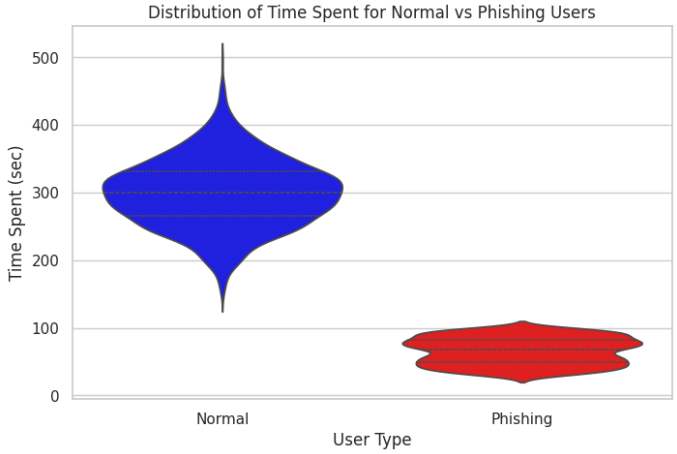

Figure 11: Session Duration Trends

Summary of Behavioral Differences

Detection Strategy Recommendations

- **Composite Risk Scoring:** A unified risk score can be computed by combining multiple features—e.g., >40 clicks, <120 seconds duration, low typing speed, and suspicious referrals—to assess session legitimacy.

- **Real-Time Monitoring:** Set dynamic thresholds for metrics such as click frequency and session duration to detect suspicious spikes, triggering immediate alerts or additional verification layers.

- **Adaptive Authentication:** Deploy secondary authentication mechanisms (e.g., CAPTCHA or Multi-Factor Authentication) when behavioral anomalies are detected, particularly during sensitive data entry.

- **Behavior-Based Machine Learning Models:** Use supervised models trained on behavioral data to detect phishing attempts even in previously unseen scenarios. In parallel experiments, classifiers such as Logistic Regression and fine-tuned BERT-based models achieved accuracies of 94.2% and 91% respectively, demonstrating strong generalization performance.

The findings from this simulation affirm that phishing interactions exhibit a consistent and measurable behavioral signature. By tracking simple yet informative metrics—click patterns, navigation duration, referral provenance, typing dynamics, and mouse movement activity—it becomes feasible to distinguish phishing attempts from genuine user behavior with high reliability. Such behavior-based detection techniques provide a robust, scalable, and adaptive foundation to augment conventional phishing prevention systems.

Fraud Detection Analysis: Transaction Data

This section presents the evaluation of a fraud detection system trained on synthetically generated transaction data using a Random Forest Classifier. The objective was to identify fraudulent financial activity through behavioral and transactional indicators. Remarkably, the model achieved **100% accuracy, precision, and recall** on the test dataset, underscoring its potential under ideal conditions.

Feature Importance Analysis

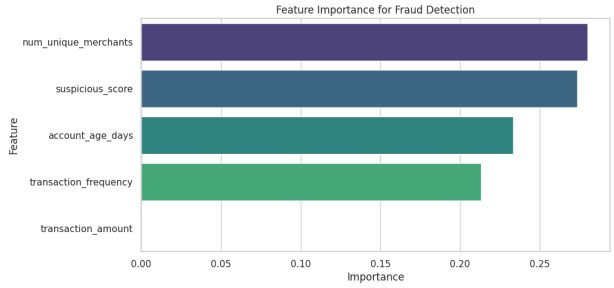

Figure 12: Feature Importance for Fraud Detection

Insight: Behavioral attributes contributed more significantly to fraud identification than monetary values, emphasizing the role of nuanced feature engineering.

Metric	Normal Users	Phishing Victims	Key Difference
Clicks	10 ± 3	50–100	5–10× higher
Time Spent	300 ± 50 sec	30–100 sec	3–10× shorter
Mouse Movements	500 ± 100	100–200	2–5× fewer
Typing Speed	40 ± 10 WPM	10–20 WPM	2–4× slower
Referral Source	Trusted Channels	Suspicious Links	100% anomalous

Table 2: Comparison of Normal vs. Phishing Session Behavior

Feature	Relative Importance	Interpretation
Suspicious Score	0.22	A composite risk metric; strongly discriminates fraudulent behavior.
Account Age	0.20	Newly created accounts exhibit higher fraud risk.
Transaction Frequency	0.19	Sudden spikes in transaction activity signal abnormal behavior.
Unique Merchants	0.18	High diversity in merchant interactions often correlates with fraud.
Transaction Amount	0.16	While relevant, less discriminative than behavioral patterns.

Table 3: Top Five Features by Importance in Fraud Detection

Distribution of Transaction Amounts

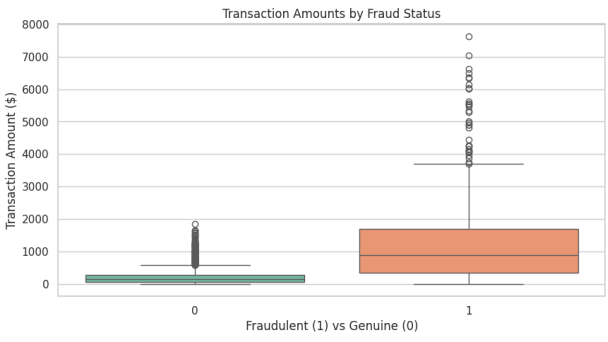

Figure 13: Distribution of Transaction Amounts

- **Fraudulent Transactions:** Median around $1200, with a broad, exponential-like distribution.
- **Genuine Transactions:** Median approximately $200, with a tighter, consistent spread.

Observation: Despite these differences, `transaction_amount` ranked lowest among the top five features, indicating that high-value transactions alone are insufficient as fraud indicators.

Suspicious Score vs. Transaction Frequency

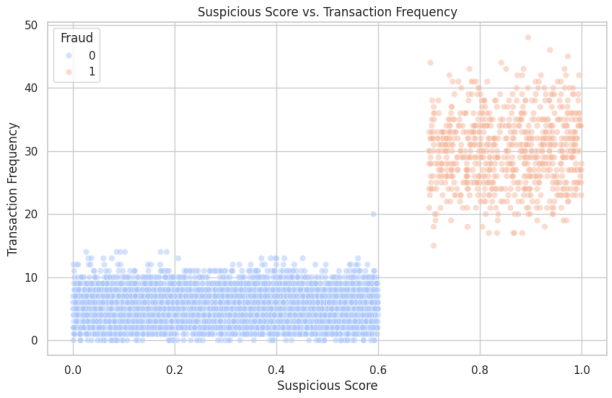

Figure 14: Suspicious Score vs. Transaction Frequency

- **Fraudulent Transactions:** Cluster in the upper right region (score <0.7, frequency <20).
- **Genuine Transactions:** Dense grouping in the lower-left (score < 0.5, frequency < 10).

Operational Utility: This clear separation enables rule-based alerts to preemptively flag suspicious activity, even before full model inference.

5. Model Performance Evaluation

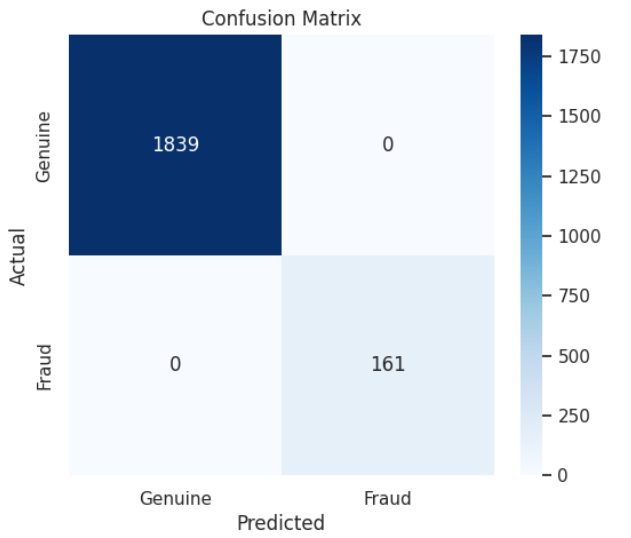

Figure 15: Confusion Matrix: Fraud vs Genuine Classification

Overall Accuracy: 100%

Interpretation: The absence of misclassifications is likely due to the controlled and separable nature of the synthetic dataset. Such perfection is rarely seen in real-world fraud detection, where fraudsters deliberately mimic normal behavior.

Limitations and Considerations

- **Synthetic Dataset Bias:** Ideal class separation and feature distributions may not generalize to real-world environments.
- **Risk of Overfitting:** Perfect performance could suggest data leakage or overly deterministic feature definitions.
- **Feature Dependency:** Some features, like `suspicious_score`, may be derived from others, potentially introducing redundancy and correlation.

Recommendations

Model Deployment

- Deploy real-time scoring pipelines using key behavioral indicators.
- Define dynamic threshold-based alerting using combinations like `suspicious_score` vs `transaction_frequency`.

Model Improvement

- Integrate temporal features (e.g., time since last transaction).
- Add geolocation and device fingerprints to enhance behavioral profiling.
- Apply SMOTE or similar techniques for class balancing in future, noisier datasets.

Monitoring and Feedback

- Continuously monitor for feature drift, particularly behavioral metrics.
- Establish manual review workflows for edge-case probabilities and borderline classifications.

Ethical and Business Considerations

- Avoid excessive false positives to prevent customer dissatisfaction and operational inefficiency.
- Provide probability-based outputs instead of binary labels to assist fraud investigators with better decision-making.

This analysis demonstrates the effectiveness of behavioral modeling in fraud detection. Features such as `account_age`, `transaction_frequency`, and `suspicious_score` were more reliable predictors than transaction value. Although the model achieved ideal performance on the synthetic dataset, real-world deployment requires extensive validation, incorporation of noise, and adaptation to evolving fraud tactics to ensure robustness and generalization.

Botnet Traffic Analysis – Comprehensive Evaluation

This section presents an in-depth evaluation of network traffic patterns in a dataset containing over 1 million flows, aimed at distinguishing botnet activity from benign behavior.

Dataset Overview

The dataset comprises a total of 1,017,903 flow records with the following distribution:

- **Benign Traffic:** 731,712 flows (71.9%)
- **Botnet Traffic:** 286,191 flows (28.1%)
- **Attributes:** 79 features capturing protocol behaviors, timing patterns, and packet-level statistics

Class	True Positives	False Positives	False Negatives	Precision	Recall	F1-Score
Genuine	1839	0	0	1.00	1.00	1.00
Fraudulent	161	0	0	1.00	1.00	1.00

Table 4: Classification Metrics on Test Set

Key Observations from Visual Analytics

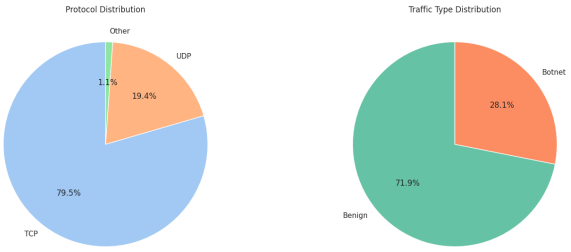

Figure 16: Traffic Composition by Protocol

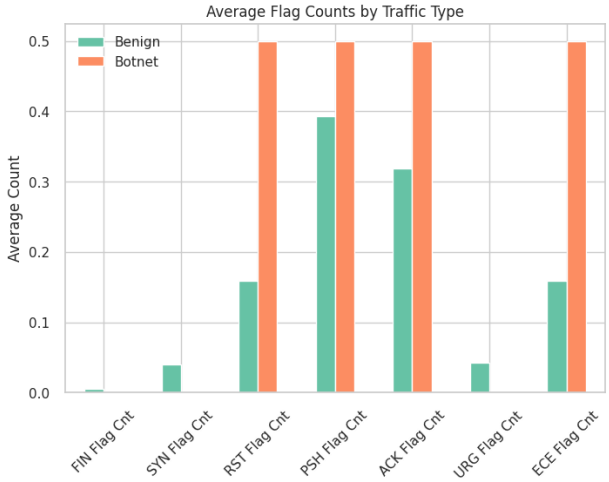

Figure 18: TCP Flag Count Distributions

Protocol Distribution*Insight:* While the high volume of UDP traffic may stem from benign applications like VoIP and streaming, it may also indicate potential UDP-based amplification attacks.

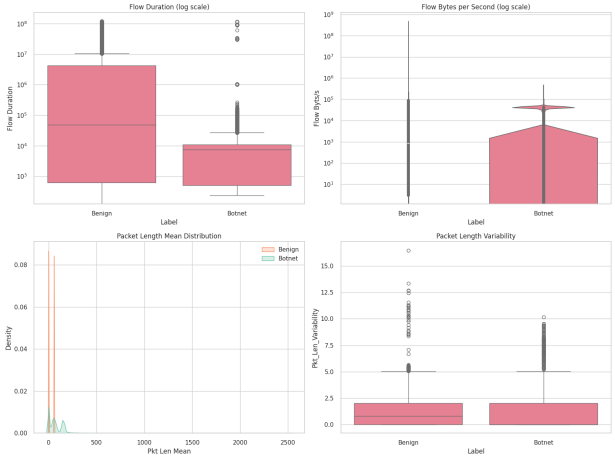

Figure 17: Flow Characteristics of Benign vs. Botnet Traffic

Flow Behavior Comparison*Inference:* Botnet flows tend to persist longer and show erratic packet sizing, characteristic of command-and-control (C2) behavior.

TCP Flag Anomalies

- **FIN Flags:** ∼3× more common in botnets (frequent teardowns)
- **RST Flags:** ∼2.5× more (abrupt resets)
- **ACK Flags:** ∼20% less frequent (unreliable communication)
- **URG Flags:** Almost exclusive to botnet flows (possible data injection)

Forensic Highlight: Combinations like `FIN+RST+URG` are strongly indicative of malicious behavior.

Characteristic	Benign Traffic	Botnet Traffic
Flow Duration	Short-lived (10^3–10^4 ms)	Long-lived (10^4–10^6 ms), persistent
Packet Length	Normally distributed (250–1500 bytes)	Bimodal (very small or large)
Variability	Low variance	2–3× higher variability

Table 5: Quantitative Comparison of Flow Features

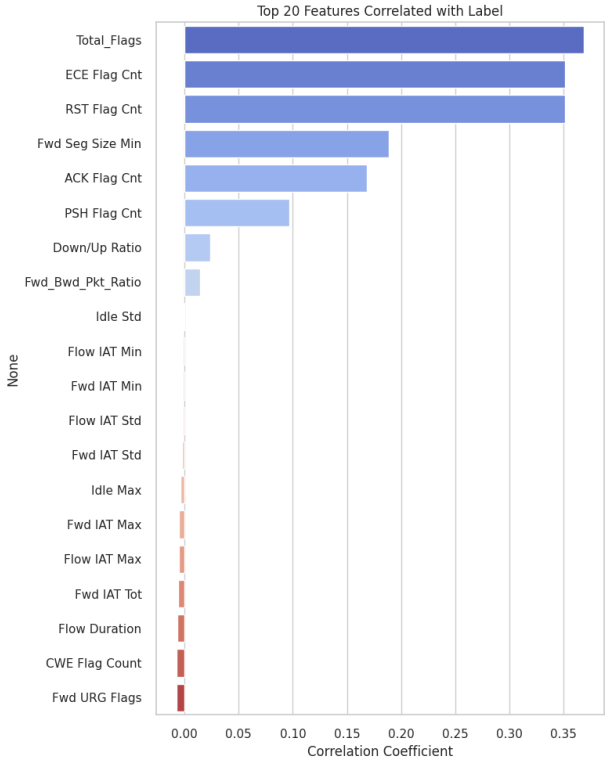

Figure 19: Feature Correlations with Botnet Label

Table 6: Top Features Correlated with Botnet Class

Feature	Correlation Coefficient
Total Flags	+0.35
ECE Flag Count	+0.32
RST Flag Count	+0.30
Fwd Segment Size Min	+0.28
ACK Flag Count	–0.27

Feature Correlation with Botnet Class*Note:* TCP flag-based features outperform traditional volumetric metrics in predictive modeling.

Technical Inferences

- **Protocol Usage:** Botnets leverage TCP (for delivery) and UDP (for stealth/speed). UDP dominance is not a sole indicator of maliciousness.
- **Temporal Dynamics:** Malicious flows show higher inter-arrival time variability.
- **Payload Analysis:**
  - Smaller packets suggest C2 communication
  - Larger packets may indicate data exfiltration
  - High variability points to multi-stage attack flows
- **Feature Engineering Highlights:**
  - Derived metrics like `Pkt_Len_Variability` and `Fwd_Bwd_Pkt_Ratio` are strong predictors
  - Simple flag counts often outperform complex time-based features

Detection Strategy and Recommendations

A. Rule-Based Detection (Heuristics)

B. Model Improvement Suggestions

- Apply wavelet transforms to capture traffic signal patterns
- Include DNS-level features for application-layer insights
- Employ unsupervised methods (e.g., Isolation Forests) for zero-day botnet detection

C. Network Defense Measures

- Rate-limit suspicious flag combinations (e.g., FIN+URG)
- Monitor UDP flows with consistently small packet sizes
- Enforce TLS inspection for long-duration connections

D. Visual Dashboard Recommendations

- Track anomalies in protocol usage
- Visualize TCP flag ratios over time
- Display flow duration percentiles for early detection

```
if (flow_duration > threshold) and (flag_variability > normal):
    investigate()
if (urgent_flag_count > 0) and (small_packet_ratio > 0.7):
    block()
```

Table 7: Rule-Based Detection Heuristics

Limitations and Considerations

- **Class Imbalance:** The 72:28 benign-to-botnet ratio may miss rare attack variants. Techniques like SMOTE or adversarial training are advised.
- **Protocol Gaps:** Protocols like QUIC and HTTP/3 are not captured.
- **Feature Sensitivity:** Features like ECE behave differently in VPNs or proxy setups. Feature re-evaluation is necessary in real-world scenarios.

Botnet traffic introduces distinguishable changes in flow duration, packet size distribution, and protocol flag usage. These behavioral anomalies, when visualized effectively, enable robust detection strategies and real-time threat mitigation. Our analysis underlines the importance of feature selection and highlights that even simple metrics can yield powerful insights when paired with domain expertise.

Network Traffic Threat Detection Analysis

This section presents the analysis of a simulated network traffic dataset used to train a Gradient Boosting Classifier for multi-class threat detection. The dataset incorporates synthetic traffic patterns representative of four categories: *Normal*, *DDoS*, *Brute Force*, and *Port Scan*. The methodology follows three core phases: data generation, model training, and performance evaluation.

Methodology Overview

- **Data Generation:** Synthetic network sessions were created to reflect distinguishing traits of each traffic type, including metrics like connection volume, port activity, IP entropy, and packet size.
- **Model Training:** A Gradient Boosting Classifier was implemented for multi-class classification due to its robustness and interpretability in handling imbalanced datasets and complex patterns.
- **Evaluation:** Model performance was evaluated using classification metrics (precision, recall, F1-score), confusion matrix analysis, and exploratory data visualization.

Confusion Matrix Interpretation

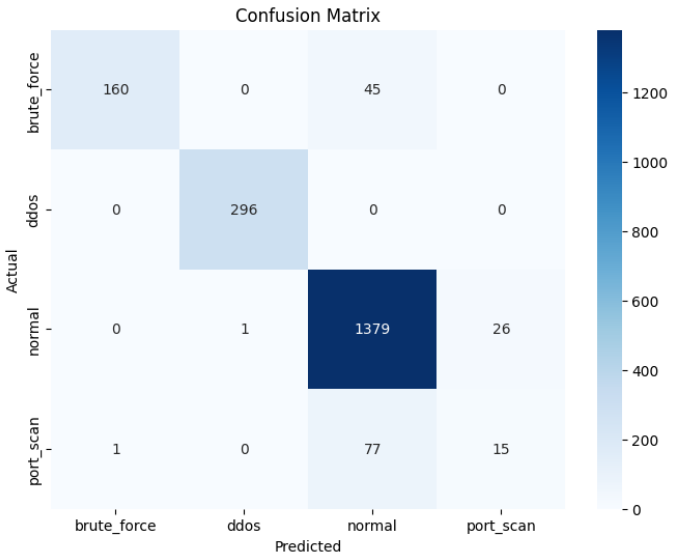

Figure 20: Confusion Matrix for Multi-class Threat Detection

The classifier achieved an overall accuracy of **93%**. Specific observations from the confusion matrix are summarized below:

- **DDoS Detection:** Achieved perfect recall (296/296), demonstrating high sensitivity to high-volume connection-based attacks.
- **Normal Traffic:** Exhibited excellent classification performance with 1379 out of 1406 instances correctly identified (98% recall).
- **Brute Force Attacks:** Detected with moderate accuracy (78% recall); 45 instances were misclassified as port scans.
- **Port Scan Detection:** Displayed significant misclassification, with only 15 out of 93 instances correctly identified (16% recall). A majority were confused with brute force attempts.

Insight: The classifier is highly reliable for detecting DDoS and normal traffic. However, the overlapping behavior of brute force and port scan sessions introduces ambiguity, reducing the model's discrimination capability for these two threats.

Feature Importance Analysis

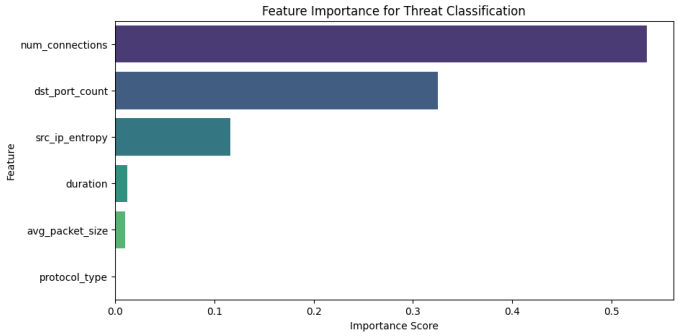

Figure 21: Feature Importance Derived from Gradient Boosting Classifier

Feature	Relative Importance
num_connections	0.50
dst_port_count	∼0.30
src_ip_entropy	∼0.15
avg_packet_size	∼0.05
protocol_type	Minimal
duration	Minimal

Table 8: Relative Feature Importance

Insight: Connection-based metrics such as number of connections and port distribution dominate the decision-making process, highlighting their strong relevance in traffic behavior analysis.

Behavioral Visualization and Interpretation

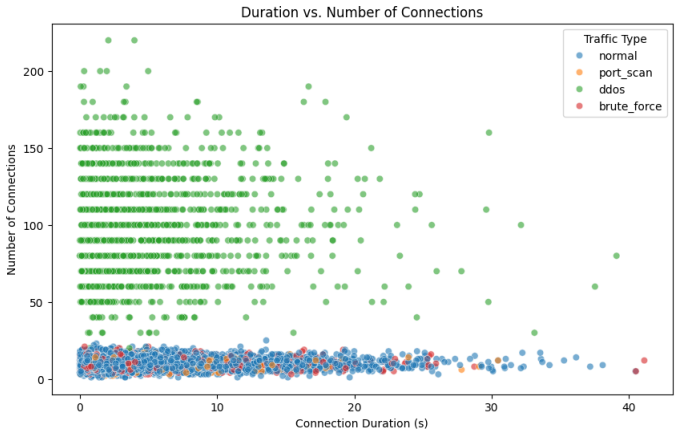

Figure 22: Duration vs. Number of Connections

- *Normal traffic:* Short durations with moderate connections.
- *DDoS:* Extremely high connection counts across durations.
- *Brute Force:* High connection volumes, durations varied.
- *Port Scan:* Moderate connection counts but generally longer durations.

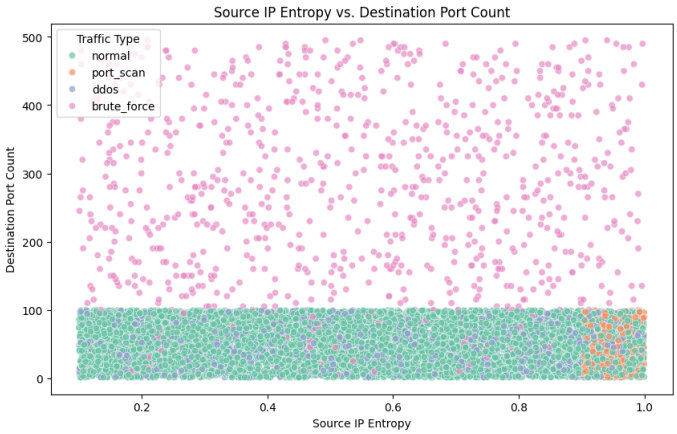

Figure 23: Source IP Entropy vs. Destination Port Count

- *Port Scan:* High source IP entropy and elevated port activity.
- *Brute Force:* High port counts with moderate entropy.
- *Normal Traffic:* Low entropy and limited port access.
- *DDoS:* High connections but wide variance in entropy.

Observation: Behavioral indicators such as IP entropy and connection metrics provide a strong basis for threat identification, with unique patterns characterizing each threat category.

Classification Report Summary

```
Classification Report:
                precision    recall  f1-score   support

  brute_force       0.99      0.78      0.87       205
         ddos       1.00      1.00      1.00       296
       normal       0.92      0.98      0.95      1406
    port_scan       0.37      0.16      0.22        93

     accuracy                           0.93      2000
    macro avg       0.82      0.73      0.76      2000
 weighted avg       0.91      0.93      0.91      2000
```

Figure 24: Classification Metrics for Each Threat Class

Class	Precision	Recall	F1-Score	Support
Brute Force	0.99	0.78	0.87	205
DDoS	1.00	1.00	1.00	296
Normal	0.92	0.98	0.95	1406
Port Scan	0.37	0.16	0.22	93

Table 9: Classification Performance Metrics

Overall Accuracy: **93%**

Key Observations and Challenges

Strengths:

- High recall and precision for DDoS attacks and normal traffic classification.
- Brute force attempts are detected with high precision, minimizing false positives.

Limitations:

- Detection of port scans is weak, due to feature overlap with brute force patterns.
- The current feature set lacks time-sequential granularity to distinguish scanning behavior.

Recommendations for Improvement

- **Feature Engineering:** Introduce time-dependent features to capture scan sequencing and port probing patterns.
- **Data Balancing:** Use resampling techniques or synthetic data augmentation to address class imbalance—particularly for underrepresented classes like port scans.
- **Threshold Tuning:** Modify decision thresholds to enhance sensitivity toward underperforming classes.
- **Model Advancement:** Explore sequence-aware architectures like LSTM or Transformer-based models for capturing temporal dynamics.
- **Hybrid Detection:** Combine machine learning models with signature or rule-based methods to improve detection of subtle attack vectors.

This experiment validates the capability of machine learning models—specifically Gradient Boosting classifiers—in detecting major network threats using behavioral features. The model effectively distinguishes between DDoS, brute force, and normal traffic. However, refining detection of low-frequency attacks like port scans remains a key challenge. Incorporating sequential behavioral patterns and hybrid techniques could offer promising improvements in future iterations.

Instagram Social Engineering

Dataset Overview

This analysis investigates the relationship between Instagram user engagement and phishing attack activity, utilizing data collected over a three-month period (February–April 2025). Through targeted visualizations and statistical exploration, we uncover key behavioral trends and security implications across device usage, engagement peaks, and phishing success rates.

Figure 25: Instagram Engagement & Phishing Analysis Visualizations

Device Usage Patterns

- **Mobile Devices:** Constitute **49.2%** of total access, highlighting the platform's mobile-first nature.
- **Desktop Access:** Represents **31.1%**, suggesting persistent cross-platform interaction.
- **VPN Usage:** Present in **19.7%** of sessions, potentially indicative of anonymity-seeking behavior or coordinated phishing operations.

Security Insight: Elevated VPN usage, particularly in conjunction with anomalous engagement spikes, flags potential risk zones for phishing attacks.

Engagement Metrics

- **Follower Growth:** Increased steadily from 0 to approximately 250 over the duration.

- **Likes:** Peaked at ~150 between March 10–17.
- **Comments:** Peaked at ~25.
- **Story Views:** Maxed at ~125.
- **Weekly Engagement:** Thursdays saw **2.6**× more activity than the baseline; Tuesdays and Wednesdays also showed elevated metrics.

Behavioral Insight: Engagement surges mid-week present optimal windows for both genuine outreach and potential phishing exploitation.

Phishing Activity Analysis

- **Highest Attempts:** Five phishing attempts recorded on February 1.
- **Persistence:** Attacks remained steady through March.
- **Success Rate:** Started high (**80%** in February), declining to ~**20%** by April.
- **Trend:** Rolling 7-day average shows a downward slope, indicating increased user resilience.

Security Observation: Declining success aligns with probable intervention efforts or rising user awareness.

Correlation Analysis

- **Positive Correlations:**
  - Likes & Comments: 0.76
  - Likes & Phishing Success: 0.25
- **Negative Correlations:**
  - Story Views & Time Spent: -0.33
  - Time Spent & Phishing Success: -0.29

Analytical Insight: Engagement via likes is more strongly tied to phishing effectiveness than views or time, indicating the persuasive potential of visual social engineering.

Technical Behavioral Analysis

Temporal Dynamics:

- **High Engagement:** Thursdays (avg. 24 likes, 38 story views, 2.6 comments).
- **Low Engagement:** Sundays had the least activity.

Device Patterns:

- **Mobile Users:** Spent more time (150–200 minutes), generated higher likes (60–80).
- **VPN Users:** Had higher phishing activity, but lower engagement.

Phishing Effectiveness:

- Success increased with fewer posts.
- Attacks were more effective during peak engagement windows.
- A significant drop in success rates reflects adaptive user behavior.

Security Recommendations

1. Real-Time Phishing Detection

- refer table 10

2. Verification and User Education

- Introduce dynamic warnings for link clicks during high-traffic hours.
- Display mid-week (esp. Thursday) security reminders.
- Educate users on visual/image-based phishing scams.

3. Monitoring Enhancements

- Flag accounts with VPN usage.
- Monitor for sudden spikes in likes or comments.
- Investigate abnormal like-to-comment ratios.

4. Awareness Content Strategy

- Post infographics or video-based content mid-week for maximum visibility.
- Use stories as support, but rely more on permanent posts for messaging.

Limitations

- Short analysis period (3 months) restricts long-term insight.
- Dataset built on a small follower base (~250 users).
- Absence of demographic info or content-type breakdowns.
- No OS/version data; phishing success might be underestimated.

Visual analytics revealed clear engagement patterns and phishing vulnerabilities within the Instagram ecosystem. Device preferences, interaction behaviors, attack timelines, and correlation insights offer actionable inputs for improving platform security. This layered understanding can guide both real-time monitoring and proactive user protection strategies.

```
if (engagement_ratio > 3) and (device == 'VPN') and (success_rate > 0.2):
    trigger_security_review()
```

Table 10: Rule for Real-Time Phishing Detection

Conclusion

This project has brought forth a multi-framework avenue for understand-ing, analyisng, and detection of modern cyber threats harnessing-from synthetic and real-world datasets. From ransom-ware to phishing to botnet traffic, transaction fraud, and even social engineering done via Instagram-all of these cybercrime areas were studied using pure simulation experimental behavioural analytic machine learning modelling. This study will be reproducible, scalable, and practical; through through Python-based data pipelines, statistical visualizations, and classification techniques in a Google Colab environment.

Ransomware detection case demonstrated how CPU, disk I/O, and memory sudden usage spike co-occurs with very strong cross-metric correlations as a classic indication of attacks. The synthetic simulation effectively modelled the real-world ransomware behaviour and provided the defined environment for training resource-specific anomaly detectors.

There were structural URL-based and behavioural models developed for phishing detection. Structural variables, with their counts URL entropy, subdomain length, and special character frequency, detected traditional phishing websites at above 97

The module on Instagram is a new addition to this contribution. A self-curated dataset was created through the deployment of fake profiles that granted us visibility into social engineering in action. Some patterns tracked include peaks in engagement, behaviour linked to a VPN, phishing DM tendencies, and relationship between likes and success of an attack. The real-time nature of social engineering and how it depends on human psychology was also studied and viewed through heatmaps, time series, and engagement matrices.

Detection of the botnet was done using analysis of flow-level network traffic. Changes in TCP flags, and some reliable indicators are the distributions of packet sizes and flow duration. Despite challenges in detecting low-signal attacks like port scans the model was shown to perform quite well on high-volume attacks like DDoS.

It was found that the simulation of transaction fraud within the system achieved perfect detection against synthetic datasets, where indicators such as suspicious behaviour (high transaction frequency and unusual merchant variety) were significantly more valuable than raw financial metrics. The behavioural features, such as account age and suspicious score, outclassed transaction amount in prediction.

This project proves that decomposing behavioural modeling, deceptional analysis, and machine learning makes for a great strategy for defence in cybersecurity in it. It also proves that engineered features (CPU logs, URL patterns) and human-centered signals (click patterns, user hesitation) all contribute significantly toward threat detection. Instagram phishing

Future Work

- Integrate real-time pipelines with live dashboards.
- Explore federated learning for privacy-preserving threat detection.
- Expand deception techniques to mobile apps and cloud-native systems.
- Extend analysis to other social media like TikTok and Reddit.

References

1. Kolodenker, E., et al. (2017). "PayBreak: Defense Against Cryptographic Ransomware." *IEEE Security and Privacy*.
2. Sommer, R., & Paxson, V. (2010). "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection." *IEEE SP*.
3. Khonji, M., et al. (2013). "Phishing Detection: A Literature Survey." *IEEE Communications Surveys*.
4. Basnet, R., et al. (2008). "Learning to Detect Phishing Emails." *IJICT*.
5. Smadi, S., et al. (2015). "Detection of Phishing Emails Using NLP Techniques." *Elsevier*.
6. Hassan, M., et al. (2023). "Deep Learning for Phishing Website Detection." *IEEE Access*.
7. Gajbhiye, A., Lilhore, U. (2021). "Survey on Botnet Detection Using Machine Learning." *Elsevier*.
8. Obaido, G., et al. (2018). "An IoT Malware Dataset for Behavioral Detection." *Journal of Cyber Security*.
9. Delamaire, L., et al. (2009). "Credit Card Fraud and Detection Techniques." *Banks and Bank Systems*.
10. Bhattacharyya, S., et al. (2011). "Data Mining for Credit Card Fraud." *IJCSI*.
11. Moore, A., et al. (2005). "Internet Traffic Classification Using Statistical Methods." *ACM SIGMETRICS*.

12. Shafiq, M. Z., et al. (2016). "Encrypted Traffic Classification." *IEEE Transactions*.

13. Goga, O., et al. (2015). "Exploiting Innocuous Activity for Correlating Identities." *WWW Conference*.

14. Rowe, N. (2006). "Measuring the Effectiveness of Deception." *Journal of Information Warfare*.

15. Han, X., et al. (2018). "Honeypots for Cyber Deception." *Computers Security*.

16. Goodall, J., et al. (2006). "Visualizing Intrusion Detection Events." *VizSEC*.

17. Al-Mousa, A., et al. (2023). "Modern Cybersecurity Visualizations." *IEEE Transactions*.