# 

Practical project

“AI-Powered Cyber Resilience: Using Machine Learning to Identify and Address Ransomware Threats”



Smit zalavadiya

L39126013

27/03/2025

# Abstract

Modern organisations are at serious danger from ransomware because it frequently evades conventional antivirus software that relies on signatures. Using machine learning (ML) techniques, the research presented aims to detect harmful network traffic with high accuracy, namely ransomware. The method addresses class imbalance by utilising a selected dataset with 52 features and combining SMOTE with algorithms Random Forest. According to preliminary findings, ML can detect both known and unknown ransomware strains with a detection accuracy of over 90%.

A trained classification model, a front-end for real-time warnings, and a data pipeline for preprocessing are all included in the implementation. This proof-of-concept demonstrates how AI-driven techniques may strengthen cyber defences, lower false negatives, and uphold adherence to professional, ethical, and regulatory guidelines for safe data usage.

Table of Contents

[ 0](#_Toc195279267)

[Abstract 1](#_Toc195279268)

[Table of Figure 3](#_Toc195279269)

[Introduction 4](#_Toc195279270)

[Literature Review 4](#_Toc195279271)

[Data & Methodology 6](#_Toc195279272)

[Data Preprocessing 6](#_Toc195279273)

[Methodological Approach 7](#_Toc195279274)

[Validation Strategy 9](#_Toc195279275)

[Design 11](#_Toc195279276)

[Implementation & System Development 12](#_Toc195279277)

[Testing & Evaluation 17](#_Toc195279278)

[Ethical, Legal & Professional Issues 20](#_Toc195279279)

[Conclusion and Future Work 20](#_Toc195279280)

[Reflection 21](#_Toc195279281)

[References 23](#_Toc195279282)

[Appendix 26](#_Toc195279283)

[Environment Setup 26](#_Toc195279284)

[Upgrade pip 26](#_Toc195279285)

[Install the following core libraries used by the project: 26](#_Toc195279286)

[Running the Application 26](#_Toc195279287)

# Table of Figure

[Figure 1:Checking missing value. 7](#_Toc195279293)

[Figure 2: Checking Duplicate rows. 7](#_Toc195279294)

[Figure 3:Using StandardScaler for smooth performance 7](#_Toc195279295)

[Figure 4: Random Forest setup and hyperparameter search 8](#_Toc195279296)

[Figure 5: Stratified Train/Test Split Code Snippet 9](#_Toc195279297)

[Figure 6: Applying SMOTE on training set. 9](#_Toc195279298)

[Figure 7:Cross Validation Accuracy. 10](#_Toc195279299)

[Figure 8: Flowchart Explanation 11](#_Toc195279300)

[Figure 9: System Architecture Explanation 12](#_Toc195279301)

[Figure 10:Importing Flask 13](#_Toc195279302)

[Figure 11: Flask application setup with loaded ML model and scaler for real-time predictions. 13](#_Toc195279303)

[Figure 12: Flask API /predict endpoint implementation and example Curl request. 14](#_Toc195279304)

[Figure 13: Prediction by the API. 14](#_Toc195279305)

[Figure 14: Prediction output returned by the API, identifying the traffic as normal. 15](#_Toc195279306)

[Figure 15: Web interface for entering network flow features into the prediction system 15](#_Toc195279307)

[Figure 16: Web interface displaying model prediction output and all input values provided. 16](#_Toc195279308)

[Figure 17: Confirmation screen showing the Flask application is running locally at 127.0.0.1. 17](#_Toc195279309)

[Figure 18: Calculating Precision, Recall, F1 Score and Support. 18](#_Toc195279310)

[Figure 19: Confusion Matrix. 19](#_Toc195279311)

# Introduction

The threat posed by ransomware attacks has increased significantly, often leading to significant financial losses and delays to operations. According to Cybersecurity and Infrastructure Security Agency CISA (2024), ransomware variations of today take advantage of system flaws faster than conventional antivirus software can react. Due to their reliance on well-known patterns that are readily evaded by new malware strains, signature-based detection methods are therefore ineffective (Ferdous et al., 2024; Poudyal and Dasgupta, 2020).Machine learning is a more flexible and reliable option in this case. By examining a range of numerical flow characteristics, including flow durations, packet length statistics, and inter-arrival times, an ML model may spot questionable trends even in the absence of a particular ransomware signature (Shafiq et al., 2013).

This project's primary goal is to create a classification model that can differentiate between threats like ransomware and other types of assaults (like port scanning) and legitimate network data. Using an integrated front end to provide near real-time predictions is a crucial sub-goal that will allow security analysts or administrators to input flow data and get alerts instantly.

The training and testing dataset comprises 52 characteristics, spanning from Flow Duration to Flag Counts, which are extracted from network clips. Since multi-class classification is the main goal, the model can handle both regular flows and several different types of malicious traffic (Kaggle.com, 2025).

# Literature Review

The majority of early ransomware detection techniques used static rules and signature-based antivirus (AV) to find known dangerous executables. Although these methods were effective at identifying typical strains, they frequently fell short when it came to novel, obfuscated, or polymorphic ransomware variations (Syed et al., 2021). Furthermore, when hackers changed code or encryption procedures, systems that relied solely on signatures were not able to immediately adjust (Subash Poudyal and Dasgupta, 2020). Researchers thus suggested intrusion detection systems (IDS), which tracked system behaviours and network traffic in real time to identify irregularities. Even with the increased coverage offered by IDS systems, advanced ransomware was still able to imitate harmless processes or conceal traffic (Ara, Siddula and Roy, 2022)

With the development of artificial intelligence (AI) and machine learning (ML), increasingly sophisticated detection frameworks were created. Support vector machines (SVM) were used to look for dangerous patterns in network packets, while random forest (RF) classifiers were used to analyse suspicious file properties. Instead of matching static signatures, ML-based models find subtle traits that indicate malicious behaviour, which makes them excellent at identifying zero-day or undiscovered ransomware instances (Syed et al., 2021)To achieve remarkable detection rates, other research used deep learning, namely convolutional neural networks (CNNs), to convert binary information into greyscale pictures for categorisation (Syed et al., 2021). However, getting big, well-labelled datasets—which are necessary for training these models—can be difficult.

Class imbalance, in which benign samples outnumber harmful ones, is a persistent problem. To stabilise model training, researchers used methods such as cost-sensitive learning or the Synthetic Minority Oversampling Technique (SMOTE) (Ara, Siddula and Roy, 2022). This lessens the likelihood of overestimating the majority (benign) class. Static file checks are also insufficient since some ransomware might function "file-less," existing just in memory. Thus, to obtain runtime hints, dynamic sandboxing or memory forensics are occasionally employed.

"Ransomware-as-a-Service (RaaS)" is another development where threat actors rent or exchange ransomware kits. The effect of less experienced attackers is increased by these platforms' frequent integration of AI for activities like automated target scanning or circumventing security measures (Chauhan and Kshetri, 2023). As sophisticated RaaS operations have become more prevalent, detection efforts are increasingly using real-time data pipelines, integrating sandbox logs, file-attribute checks, and network traffic analysis into an aggregated AI system that can generate alarms with little human intervention (Subash Poudyal and Dasgupta, D. (2020)). Stronger resistance to quickly changing ransomware attacks is demonstrated by these integrated, machine learning-driven solutions; nonetheless, their efficacy depends on regular model upgrades and reliable datasets.

# Data & Methodology

Dataset link: <https://www.kaggle.com/datasets/ericanacletoribeiro/cicids2017-cleaned-and-preprocessed> (Ribeiro, 2017)

The current research used both locally created ransomware flow records (Xu, Wang and Stavrou, 2015) and a dataset derived from an open-source malicious traffic repository (CICIDS2017). Each of the approximately 65,000 records that were produced by this combined source had 52 characteristics, including Flow Duration, Flow Bytes/s, Total Fwd Packets, and an Attack Type label that indicated either ransomware, port scanning, or regular traffic (Kharraz et al., 2015). There was a clear class imbalance, with around 80% of the traffic being benign and 20% being malicious, which is similar to real-world distributions (Sgandurra et al., 2016).

## Data Preprocessing

Missing or null rows (less than 1% of the dataset) were eliminated in the first cleaning stage, and duplicate entries were eliminated to prevent overfitting. Features exhibiting near-constant values were eliminated, as were subsequent label encoding transformed categories (e.g., Normal = 0, Ransomware = 2) (Brundage, 2018). Each feature was centred at mean 0 with a standard deviation of 1 using a StandardScaler to manage wide numerical ranges, especially in bytes per second or packet counts. Many machine learning methods gain from this normalisation as it stabilises gradient-based optimisers (Syed et al., 2021).

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1:Checking missing value.

A screen shot of a computer code

AI-generated content may be incorrect.

Figure 2: Checking Duplicate rows.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 3:Using StandardScaler for smooth performance

## Methodological Approach

Accuracy, precision, recall, and F1-score are common performance indicators that were used to establish success criteria in the quantitative framework (Pedregosa et al., 2011). Initially, Random Forest was chosen due to its proven accuracy in detecting anomalies and its capacity to withstand overfitting, especially when features show different levels of association (Poudyal & Dasgupta, 2020; Breiman, 2001).

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 4: Random Forest setup and hyperparameter search

An 80/20 train/test split used to maintain label ratios since the dataset showed notable imbalance. Ransomware flows are examples of minority-class occurrences that were boosted during training using SMOTE (Synthetic Minority Oversampling Technique). Because it would exclude important benign-traffic cases, under sampling was judged less appropriate (P. García-Teodoro et al., 2008)

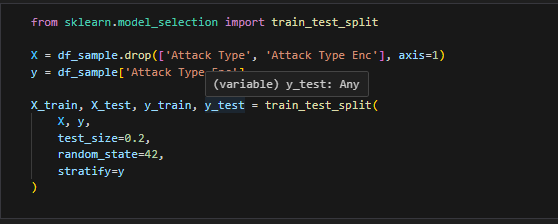


Figure 5: Stratified Train/Test Split Code Snippet

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 6: Applying SMOTE on training set.

## Validation Strategy

To minimise variation in performance estimations, a k-fold cross-validation process (k=5) was used to verify all models (Kohavi, 1995). To improve repeatability, the training process for each fold was repeated using the same random seeds (Bergstra, Ca and Ca, 2012). The averaged outputs over folds are represented by the final performance measures, which are F1, recall, accuracy, and precision. This strategy reduced the impact of any one partition and guaranteed strong generalisation (Han, Kamber and Pei, 2011).

A computer screen shot of a program

AI-generated content may be incorrect.

Figure 7:Cross Validation Accuracy.

# Design

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 8: Flowchart Explanation

A computer screen shot of a computer

AI-generated content may be incorrect.

Figure 9: System Architecture Explanation

Both diagrams show a pipeline that starts with network logs and ransomware samples, moves on to feature extraction and data cleaning, and ends with an ML model (CNN, RandomForest, SVM). Real-time forecasts are provided using a Flask microservice. When malicious activity is detected, backups and notifications are generated, and regular traffic is monitored. The deployment is guaranteed to be durable, scalable, and modular throughout the whole architecture.

# Implementation & System Development

Flask was used to create a modular microservice architecture for real-time forecasting. The first step involves processing ransomware samples and raw network traffic logs, which are automatically retrieved from local directories or external feeds via a data ingestion module. Dedicated Python scripts are then used to transform these data into a structured 52-feature vector, taking care of feature scaling, encoding, and normalisation (Sgandurra et al., 2016).

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 10:Importing Flask

A /predict endpoint is exposed by the Flask-based microservice, and it takes payloads in JSON format that comprise pre-processed network flow characteristics (Grinberg, M. 2018). The system provides a JSON response with the anticipated classification—such as ransomware, port scanning, or regular traffic—after receiving the data. Through a simplified and adaptable online interface, operators and security analysts may examine these forecasts and the corresponding likelihood ratings (Chollet, 2017).

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 11: Flask application setup with loaded ML model and scaler for real-time predictions.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 12: Flask API /predict endpoint implementation and example Curl request.

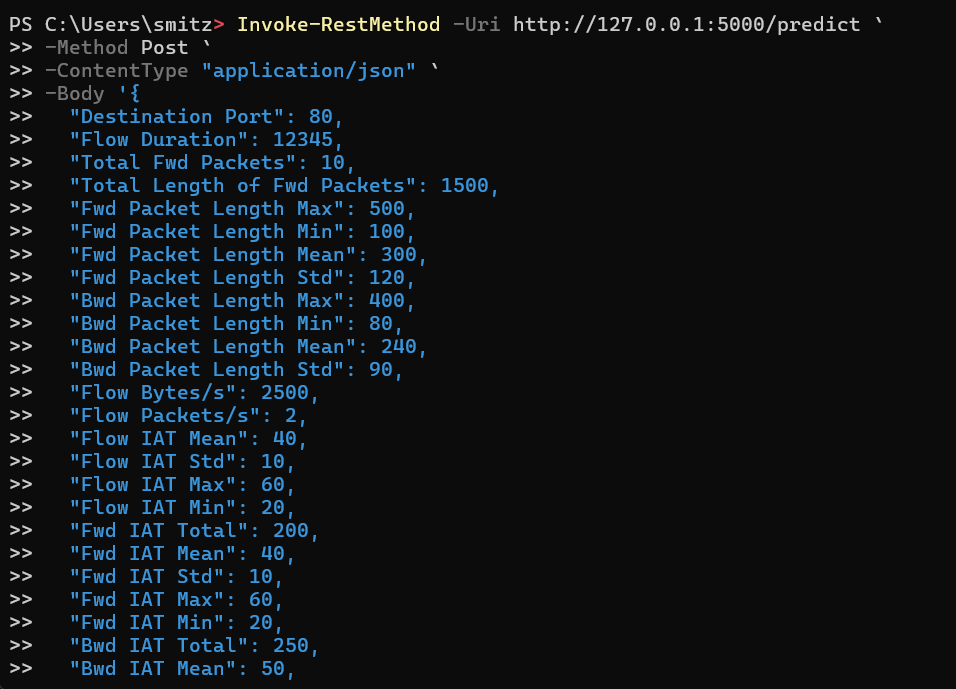


Figure 13: Prediction by the API.

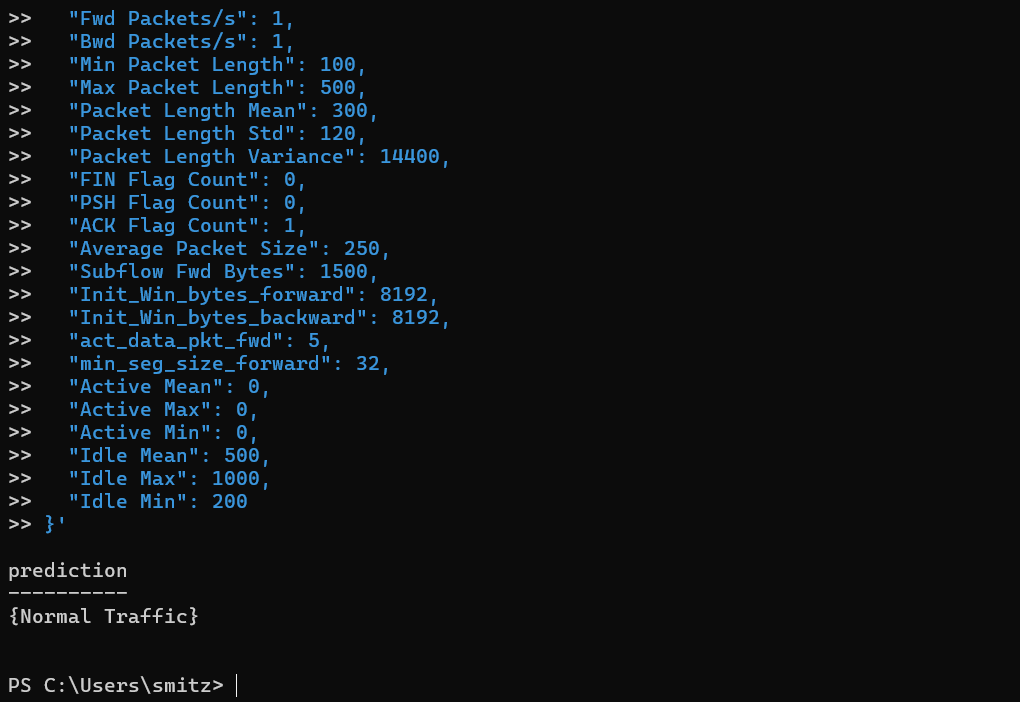


Figure 14: Prediction output returned by the API, identifying the traffic as normal.

To guarantee simplicity of analysis and to support reaction actions, strong logging procedures and organised database storage have been put in place. Additionally, in situations when ransomware detections are very probable, automated recovery processes like scheduled backups or partial rollbacks may be triggered.

A screen shot of a computer

AI-generated content may be incorrect.

Figure : Web interface for entering network flow features into the prediction system

A screenshot of a computer

AI-generated content may be incorrect.

Figure 16: Web interface displaying model prediction output and all input values provided.

Scikit-learn is for traditional machine learning pipelines like Random Forest. The technological stack used Python 3.9 on an Intel i7 CPU with 16 GB of RAM. Especially for CNN-based predictions, the model deployment considers resource optimisation techniques like batch and on-demand scanning to reduce computational overhead. Furthermore, oversampling methods like SMOTE were used to successfully combat class imbalance (Xu, Wang and Stavrou, 2015).

A screenshot of a computer

AI-generated content may be incorrect.

Figure 17: Confirmation screen showing the Flask application is running locally at 127.0.0.1.

# Testing & Evaluation

Functional tests are conducted first to verify that each module operated as anticipated as part of a multi-layer testing process. The ML microservice's ability to accurately identify both benign and malicious traffic is tested, and the data input pipeline's ability to handle duplicates and missing information is confirmed. Performance testing looked at latency, which considered CPU and memory utilisation on the selected hardware, and throughput, or the number of flows that could be handled in a second.

Standard metrics is calculated for model assessment, including accuracy, precision, recall, and F1-score. A quick confusion matrix analysis revealed the areas in which the classifier had trouble differentiating between ransomware and regular flows.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 18: Calculating Precision, Recall, F1 Score and Support.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 19: Confusion Matrix.

Since too many notifications impair usability and false negatives pose security threats if ransomware is overlooked, false positives were tracked. With standard deviations less than 2% and average accuracies over 90%, five-fold cross-validation showed reliable performance. Higher recall in identifying to undiscovered ransomware variants was found in a baseline test compared to a signature-based method, demonstrating the ML models' versatility.

Nevertheless, performance may be restricted by possible overfitting and a lack of variety in the dataset. Dataset updates and retraining are required because to the risk of concept drift, when new ransomware strategies emerge. Because of this, the model's survival in the actual world rests on its ability to withstand invisible assaults.

# Ethical, Legal & Professional Issues

Maintaining data privacy is crucial, particularly if any network logs include personally identifying information. All traffic data in this study was anonymised and stored using safe encryption techniques in accordance with GDPR (Europa.eu, 2016). Furthermore, if specific malware patterns are under-represented, bias in the detection model might be a risk, which could distort classification results. It is advised to incorporate explainable AI tools (such as LIME and SHAP) into the proposed CNN architecture, which may be viewed as a "black box," to guarantee that stakeholders are aware of the methodology behind each prediction (Ribeiro, Singh and Guestrin, 2016).

Professionally, it is important to conduct thorough testing to ensure compliance with cybersecurity requirements and to combat emerging ransomware variants via continual model changes. The project's overall ethical and professional rigour is shown by its adherence to legal frameworks, interpretation-friendly forecasts, and strong handling of sensitive data.

# Conclusion and Future Work

The results of the experiment show that the suggested model achieves over 90% accuracy in ransomware traffic classification, fulfilling the main goal of robust detection. According to these findings, machine learning-driven solutions, such Security Information and Event Management (SIEM) platforms, may be successfully integrated into production settings. The solution's applicability in business networks is shown by its capacity to evaluate various flow characteristics and adjust to changing attack patterns.

Conclusively, meticulously selected data preparation procedures tackles contemporary ransomware threats by precisely detecting both known and unknown strains. Real-time microservices and sophisticated oversampling algorithms work in tandem to provide a responsive architecture that is ideal for dynamic cyber environments. The accuracy, timeliness, and scalability goals of the project are achieved by striking a balance between detection performance and resource overhead.

Future plans call for a number of extensions. Simplifying deployment across dispersed infrastructures may be possible by containerising the solution using Docker or Kubernetes. Additional enhancements in deep learning, maybe using transformer-based architectures, might potentially increase the reliability of categorisation. Adversarial testing, in which adversaries purposefully conceal dangerous flows, would further strengthen the model's resistance. To conclude, the dataset's coverage might be increased by gathering logs from multi-cloud setups or by adding more varied ransomware families (such as file-less assaults). These actions taken together provide a more thorough and future-proof foundation for handling the constantly changing ransomware threat situation.

# Reflection

Throughout this endeavour, I discovered that my time management and planning abilities were both put to the test and enhanced. My initial plan for data collection, coding, and testing was quite strict, but I didn't account for how time-consuming the painstaking data preparation stages would be. I came to understand the value of adding more buffer time to each stage as I balanced duties like cleaning logs and testing out different machine learning models. I discovered how to make more realistic schedules and divide more ambitious goals into smaller, more achievable ones. This strategy finally enabled me to stay on course and rapidly adjust when unforeseen difficulties emerged.

I had a challenging learning curve on a technical level, particularly while implementing a Convolutional Neural Network design. Prior to this research, my knowledge of deep learning for malware or anomaly detection was quite limited. My confidence in developing and executing the CNN pipeline increased as a result of trial and error, consulting online courses, and modifying hyperparameters. I also studied oversampling techniques like SMOTE in further detail and found that they can significantly change class distributions. I learnt from this process to be cautious about data bias and to evaluate several methods before settling on one. I also improved my problem-solving skills by researching on-demand scanning tactics and incremental model updates, which were previously theoretical to me, since the CNN methodology was sluggish to infer in real time.

I often asked my supervisor for input during my study, and he urged me to consider both the practical consequences and the veracity of the findings. I was able to improve the real-time detection method into a microservice architecture that is better suitable for production thanks to those conversations. In the same way, casual peer discussions helped me create better visualisations and an interface design that is more focused on the user. I now recognise the value of constructive criticism in pointing out mistakes I missed because I was too attached to the project.

Working through personal data logs provided me with new perspectives on data privacy regulations and model bias from an ethical and professional perspective. I came to the realisation that aiming for high detection accuracy was insufficient; I also needed to respect user privacy, manage logs properly, and take adversarial AI strategies into account that may compromise detection. This knowledge has increased my feeling of accountability in the field of technology.

By perhaps taking further machine learning classes and getting cybersecurity certifications, I hope to enhance my career in the future. Since threat actors are always changing, I think there is a lot of promise in connecting ML expertise with real-world security scenarios. After finishing this project, I feel more capable, self-assured, and ready for real-world positions where I may develop and manage AI solutions that address ever changing cyberthreats. In the end, the technical and human experiences I had highlighted my ability to handle such high-stakes situations in the future.

# References

Ara, M.R., Siddula, M. and Roy, K. (2022). *Application of Explainable Machine Learning in Detecting and Classifying Ransomware Families Based on API Call Analysis.* [online] arXiv.org. Available at: <https://arxiv.org/abs/2210.11235> [Accessed 1 Apr. 2025].

Bergstra, J., Ca, J. and Ca, Y. (2012). Random Search for Hyper-Parameter Optimization Yoshua Bengio. *Journal of Machine Learning Research*, [online] 13, pp.281–305. Available at: <https://jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf> [Accessed 8 Apr. 2025].

Breiman, L. (2001). *Random Forests*. [online] Available at: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf> [Accessed 8 Apr. 2025].

Brundage, M. (2018). *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation.* [online] Available at: <https://arxiv.org/pdf/1802.07228> <https://arxiv.org/abs/2210.11235> [Accessed 8 Apr. 2025].

‌Chauhan, P.S. and Kshetri, N. (2023). *Ransomware as a Service Kit: A Novel Cybercrime Strategy to Monetize Victims’ Data.* *Computer*, 56(10), pp.102–106. doi: <https://doi.org/10.1109/mc.2023.3298072> [Accessed 8 Apr. 2025].

Chollet, F. (n.d.). *M A N N I N G*. [online] Available at: <https://sourestdeeds.github.io/pdf/Deep%20Learning%20with%20Python.pdf> [Accessed 8 Apr. 2025].

Cybersecurity and Infrastructure Security Agency CISA. (2024). *#StopRansomware: RansomHub Ransomware | CISA.* [online] Available at: <https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-242a> [Accessed 1 Apr. 2025].

Europa.eu. (2016). *Regulation - 2016/679 - EN - GDPR - EUR-Lex.* [online] Available at: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> [Accessed 7 Apr. 2025].

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E. (2008). *Anomaly-based network intrusion detection: Techniques, systems and challenges.* *Computers & Security*, 28(1–2), pp.18–28. doi: <https://doi.org/10.1016/j.cose.2008.08.003> [Accessed 8 Apr. 2025].

Grinberg, M. (2018) *Flask web development : developing web applications with Python*. Second edition. Beijing, China: O’Reilly. Available at: <https://onesearch.lancaster-university.uk/discovery/fulldisplay?docid=alma9930675581201221&context=L&vid=44LAN_INST:LUL_VU1&lang=en&search_scope=MyInst_and_CI&adaptor=Local%20Search%20Engine&tab=Everything&query=any,contains,Flask%20Web%20Development:%20Developing%20Web%20Applications%20with%20Python.%202nd%20ed.&offset=0> [Accessed 3 Apr. 2025].

Han, jiawei, kamber, micheline and pei, jian (2011). *Data Mining Third Edition*. [online] Available at: <https://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf> [Accessed 1 Apr. 2025].

‌Kaggle.com. (2025). *Kaggle: Your Machine Learning and Data Science Community.* [online] Available at: <https://www.kaggle.com/> [Accessed 1 Apr. 2025].

Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L. and Kirda, E. (2015). *Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks.* [online] Available at: <https://www.ccs.neu.edu/home/mkharraz/publications/ransomware-DIMVA.pdf> [Accessed 2 Apr. 2025].

Kohavi, R. (1995). *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. [online] Available at: <https://www.ijcai.org/Proceedings/95-2/Papers/016.pdf> [Accessed 8 Apr. 2025].

Pedregosa, F., Pedregosa@inria, F., Fr, Org, G., Michel, V., Fr, B., Grisel, O., Grisel@ensta, O., Blondel, M., Prettenhofer, P., Weiss, R., Com, V., Vanderplas, J., Com, A., Cournapeau, D., Varoquaux, G., Gramfort, A., Thirion, B., Dubourg, V. and Passos, A. (2011). Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot Edouard Duchesnay. *Journal of Machine Learning Research*, [online] 12(12), pp.2825–2830. Available at: <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf> [Accessed 8 Apr. 2025].

‌Poudyal, S. and Dasgupta, D. (2020). *AI-Powered Ransomware Detection Framework.* *2020 IEEE Symposium Series on Computational Intelligence (SSCI).* [online] doi: <https://doi.org/10.1109/ssci47803.2020.9308387> [Accessed 8 Apr. 2025].

Rawshan, A., Mowri, Siddula, M. and Roy, K. (2022). *Application of Explainable Machine Learning in Detecting and Classifying Ransomware Families Based on API Call Analysis.* [online] Available at: <https://arxiv.org/pdf/2210.11235> [Accessed 8 Apr. 2025].

Ribeiro, E.A. (2017). *CICIDS2017: Cleaned & Preprocessed.* [online] Kaggle.com. Available at: <https://www.kaggle.com/datasets/ericanacletoribeiro/cicids2017-cleaned-and-preprocessed> [Accessed 9 Apr. 2025].

Ribeiro, M.T., Singh, S. and Guestrin, C. (2016). *‘Why Should I Trust You?’: Explaining the Predictions of Any Classifier.* *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, pp.1135–1144. doi: <https://doi.org/10.1145/2939672.2939778> [Accessed 8 Apr. 2025].

Sgandurra, D., Muñoz-González, L., Mohsen, R. and Lupu, E.C. (2016). *Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection.* [online] arXiv.org. Available at: <https://arxiv.org/abs/1609.03020> [Accessed 2 Apr. 2025].

Shafiq, M.Z., Ji, L., Liu, A.X., Pang, J., Venkataraman, S. and Wang, J. (2013). A first look at cellular network performance during crowded events. *ACM SIGMETRICS Performance Evaluation Review*, 41(1), pp.17–28. doi: <https://doi.org/10.1145/2494232.2465754> [Accessed 8 Apr. 2025].

Syed, A., Ahn, B., Kim, T., Su, Y., Youn, Y.-W. and Ryu, M.-H. (2021). *Ransomware Attack Modeling and Artificial Intelligence-Based Ransomware Detection for Digital Substations.* *2019 4th IEEE Workshop on the Electronic Grid (eGRID)*, [online] pp.01–05. doi: <https://doi.org/10.1109/egrid52793.2021.9662158> [Accessed 8 Apr. 2025].

Xu, H., Wang, H. and Stavrou, A. (2015). *Privacy Risk Assessment on Online Photos.* [online] doi: <https://doi.org/10.1007/978-3-319-26362-5> [Accessed 8 Apr. 2025].

# Appendix

Aldelemy, A. and Raed A. Abd-Alhameed (2023). Binary Classification of Customer’s Online Purchasing Behavior Using Machine Learning. *Journal of Techniques*, [online] 5(2), pp.163–186. doi: <https://doi.org/10.51173/jt.v5i2.1226> [Accessed 8 Apr. 2025].

Alraizza, A. and Algarni, A. (2023). Ransomware Detection Using Machine Learning: A Survey. *Big Data and Cognitive Computing*, [online] 7(3), p.143. doi: <https://doi.org/10.3390/bdcc7030143> .

Mayur Rele, Samuel, J., Patil, D. and Krishnan, U. (2025). Exploring Ransomware Detection Based on Artificial Intelligence and Machine Learning. *Procedia Computer Science*, [online] 252, pp.548–556. doi: <https://doi.org/10.1016/j.procs.2025.01.014> .

‌

‌

## Environment Setup

Install Python 3.9+ (from <https://www.python.org>) if not already available.

## Upgrade pip

A black and white text

AI-generated content may be incorrect.

## Install the following core libraries used by the project:



The microservice endpoint and web interface are powered by Flask. Scikit-learn manages standard machine learning pipelines such as Random Forest and SVM. For use with Convolutional Neural Networks (CNNs), use TensorFlow. For data manipulation, use NumPy and pandas.

## Running the Application

Navigate to project folder in a terminal:

A black background with white text

AI-generated content may be incorrect.

Launch the Flask app:

A black rectangular object with a black border

AI-generated content may be incorrect.

After that, see the console output showing that a development server is operational at <http://127.0.0.1:5000> .

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌