# Classification of Bitcoin Ransomware Transactions Using Random Forest: A Data Mining Approach for Blockchain Security

Ibrahiem M. M. El Emary[1,*], Anna Brzozowska[2], Łukasz Popławski[3], Paweł Dziekański[4], Jozef Glova[5]

[1]King Abdulaziz University, Kingdom of Saudi Arabia, Saudi Arabia

[2]Faculty of Management, Czestochowa University of Technology, Poland

[3]Cracow University of Economics, Cracow, Rakowicka 27, 31-510 Poland

[4]Jan Kochanowski University in Kielce, Kielce. Stefana Zeromskiego 5, Poland

[5]Technical University of Košice, Nemcovej 32, 042-00 Košice, Slovakia

## ABSTRACT

The rapid evolution of ransomware attacks necessitates robust and scalable detection mechanisms to safeguard digital assets. This study leverages the Bitcoin Ransomware Dataset, comprising 2,916,697 transactions, to evaluate the effectiveness of the Random Forest algorithm in classifying ransomware-related activities. Through comprehensive preprocessing, including feature encoding and standardization, and exploratory data analysis (EDA), the dataset is prepared for modeling. The Random Forest model achieves an overall accuracy of 99%, demonstrating exceptional performance in identifying the majority class. However, challenges persist in classifying minority classes, highlighting the impact of class imbalance. Feature importance analysis reveals that attributes such as income, weight, and length play pivotal roles in the classification process. The study underscores the potential of Random Forest for ransomware detection while emphasizing the need for advanced techniques to address class imbalance and improve minority class performance.

**Keywords** Ransomware detection, Bitcoin transactions, Random Forest, Classification, Feature importance, Class imbalance, Machine learning, Cybersecurity

## INTRODUCTION

Blockchain technology has emerged as a transformative force in the realm of digital transactions, offering unparalleled potential through its decentralized and immutable nature. Yet, tantalizing as it may be, even this bastion of security is not impervious to cybersecurity threats, notably ransomware. The current landscape is a battleground of innovation and risk mitigation, with literature teeming with insights into blockchain technology and its associated security challenges.

Among the myriad notes of caution in the blockchain symphony, the discordant threat of malleability attacks rings loudest. Studies assert this vulnerability to be more than an anomaly, as it holds the potential for cascading financial repercussions—case in point, the $2.4 million conjured away by such attacks [1]. This Achilles' heel has put the integrity of blockchain transactions under the magnifying glass, demanding robust countermeasures developed with urgency and precision.

Intersecting with this narrative of vulnerability is the promise of integrating

Submitted 25 January 2025
Accepted 30 April 2025
Published 1 June 2025

Corresponding author
Ibrahiem M. M. El Emary,
omary57@hotmail.com

Additional Information and Declarations can be found on page 166

DOI: 10.47738/jcrb.v2i2.33



**How to cite this article:** I. M. M. El Emary, A. Brzozowska, Ł. Popławski, P. Dziekański, J. Glova, "Classification of Bitcoin Ransomware Transactions Using Random Forest: A Data Mining Approach for Blockchain Security," *J. Curr. Res. Blockchain*, vol. 2, no. 2, pp. 152-168, 2025.

blockchain within broader cybersecurity frameworks. The interplay between blockchain and cybersecurity can forge an objective-based framework, an analytical lens through which financial institutions can assess their defenses [2]. The notion is not purely theoretical; smart contracts, autonomous digital agreements on blockchain, can dynamically recalibrate security protocols in real-time, creating an agile bastion against evolving threats [3].

However, the shadow of ransomware looms large. These financially-motivated cyber adversities thrive on blockchain's cloak of anonymity and its decentralized charm [4]. Add to this a dose of collusive potential, where blockchain's strength becomes its vulnerability, manipulated by powerful entities to alter data or disrupt its flow [5]. This paradox underscores the importance of relentless vigilance and the refinement of security protocols.

Still, blockchain's role in bolstering cybersecurity is undeniable, weaving resilience through its pillars of transparency and immutability, essential for sanctifying transactions and guarding sensitive intel [6]. As its adoption widens, organizations grapple with the intricacies of seamless integration into prevailing cybersecurity architectures, particularly within varied network landscapes [7]. Herein lies the rub—balancing data integrity and scalability demands innovative remedies to sustain secure, fluid blockchain operations.

Detecting ransomware transactions stands as a cornerstone for fortifying blockchain security, especially amidst the rising tide of sophisticated ransomware attacks leveraging cryptocurrencies for illicit gains. The rapidity with which ransomware encrypts and demands payment creates an urgent need for detection mechanisms that can seamlessly integrate within blockchain contexts.

Timeliness in detection emerges as the paramount challenge, for once ransomware encrypts files, the path to recovery narrows perilously, often teetering on impossibility [8]. This urgency begets a pursuit for methods that can detect ransomware activities preemptively—before damage transpires. One promising avenue is the monitoring of application programming interface (API) sequences, heralded as a veil lifter for ransomware's shadowy operations, pinpointing threats at incipient stages [9]. Such preemptive strikes are crucial within blockchain ecosystems where transactions blink by in a heartbeat, often beyond the reach of traditional oversight.

In this evolving landscape, machine learning techniques transcend traditional methods, offering a potent ally in ransomware detection. Contemporary research has unfurled the potential of machine learning models to dissect the dynamic behaviors of ransomware with remarkable precision [10], [11]. These models are the chameleons of cybersecurity, adapting to ransomware's ever-shifting tactics designed to skirt around conventional defenses research [12]. By employing machine learning, efforts to classify ransomware-specific Bitcoin transactions have crystallized, effectively spotlighting nefarious activities that once lurked in the blockchain's murky depths [13].

Adding another layer to this arsenal are entropy-based detection methods, which spotlight anomalies by gauging file entropy—subtle deviations that might signal ransomware's partial encryption artifice [14], [15]. This method shines particularly within cloud services, where remote file storage demands a deft touch beyond the reach of customary detection paradigms.

Moreover, synthesizing static feature analytics with behavioral scrutiny has emerged as a strategy to bolster detection accuracy. Delving into the imports of Portable Executable (PE) files unravels ransomware's behavioral code, offering a vista into its operational playbook, thereby broadening the detection spectrum

[16]. This layered, multifaceted approach provides a bulwark, equipping defenses to counter a myriad of ransomware vectors with agility and depth.

In the vast and turbulent sea of cryptocurrency security, the detection of ransomware transactions through Bitcoin remains a largely uncharted expanse, especially when viewed through the prism of Random Forest (RF) algorithms. Numerous methods traverse the landscape of detection, yet RF's application for isolating Bitcoin ransomware transactions languishes, hitherto overlooked. This oversight represents a conspicuous research gap, beckoning for exploration and revelation.

Machine learning, in its multifaceted glory, has long been heralded as a beacon of hope in combating fraudulent activities within Bitcoin ecosystems. The empirical triumphs of Random Forest are no exception—Chen et al. have illuminated its superior prowess, demonstrating RF's dominance over other supervised learning techniques in pinpointing Bitcoin theft with commendable precision and recall [17]. Such findings ignite the prospect of harnessing RF's capabilities to expose ransomware transactions. Nonetheless, the targeted application of RF in the realm of ransomware detection remains less treaded, calling for specialized investigation.

Further examination surfaces the contributions of Al-Haija and Alsulami, who have pioneered classification models adept at unraveling ransomware payments amidst the labyrinthine networks of Bitcoin, with decision tree-based models yielding exceptional accuracy [18]. Their work sheds light on the ripe potential of machine learning tools—yet RF remains conspicuously absent from their focus. Here lies a tantalizing opportunity to delve into how RF could elevate detection rates when applied to ransomware's shadowed ledger.

Widening the lens, Nayyer's research underscores the critical role advanced machine learning, including ensemble methods, plays in securing Bitcoin transactions, positing them as stalwart solutions against fraud research [19]. Though ransomware detection sits peripherally in this study, it firmly plants the notion that machine learning is instrumental in safeguarding Bitcoin's integrity. Nestled within this context, RF emerges not only as a promising candidate but as an anticipatory solution with the potential for broad applicabilitys.

Given the chameleonic nature of ransomware, constantly morphing its tactics to evade detection, there arises a compelling imperative for innovative strategies. RF, with its inherent adaptability, could become an invaluable asset in preempting and neutralizing these nascent threats. As ransomware strategies evolve, so too must our defenses, requiring robust systems capable of continuous adaptation.

This study aspires to illuminate the field of blockchain security by applying the Random Forest (RF) algorithm to the classification of Bitcoin ransomware transactions, thereby bolstering existing security protocols. As current literature calmly sidesteps the niche of RF's potential within the landscape of ransomware detection on Bitcoin, this endeavor seeks to correct that oversight.

The allure of Random Forest stems from its ubiquitous success across diverse classification applications, particularly within cybersecurity realms. Prior investigations solidify RF’s standing as a versatile, high-performing model, adept at discerning fraudulent activities within Bitcoin ambitiously [18]. This study, therefore, harnesses RF's computational prowess to forge a model capable of meticulously classifying transactions, distinguishing between benign exchanges and those tinged with the specter of ransomware.

By intertwining machine learning techniques such as RF with blockchain frameworks, an innovative beacon of security emerges. This fusion promises

advancements in detection capabilities, facilitating the recognition of anomalous transaction patterns that may flag ransomware activities [20]. Thus, this research intends to add a formidable thread to the interdisciplinary fabric, shedding light on RF's utilization to fortify blockchain defenses against ransomware incursions.

Further compounding the importance of this study is the persistent evolution of ransomware tactics, necessitating adaptive security mechanisms. Current discussions accentuate the need for frameworks that can morph in response to new attack vectors [21]. By embracing RF, this analysis aims not only to enhance classification accuracy but also to engender a detection model that is resilient amidst an ever-shifting malware landscape, fortifying the blockchain's resilience.

The present study attains a grounded footing on the shoulders of research which have adeptly wielded machine learning for detecting ransomware within Bitcoin exchanges [13]. A nuanced focus on RF uncovers its unique perks—such as deftly managing voluminous datasets and its inherent resistance to overfitting—making it a fitting choice for navigating the complex maze of ransomware-laden transaction patterns.

## Literature Review

### Existing Methods for Ransomware Detection

In the sprawling battleground of cybersecurity, where ransomware wields encryption as its weapon of choice, various techniques have been forged to detect this insidious threat. Commanding a leading role in this arsenal are machine learning algorithms and statistical approaches, each bringing unique strengths to the fray against ransomware's growing sophistication.

A cornerstone of these methodologies is entropy-based detection, which probes the randomness or disorder within files to spotlight potential ransomware infections. Lee et al. have championed this approach, demonstrating that even partially encrypted ransomware can be unmasked by scrutinizing file entropy, especially when assessed post-decoding of base64 encodings [22]. This method earns its stripes by unearthing anomalies in file architecture that betray malicious tampering. Yet, as Davies et al. caution, ransomware architects have not been idle, often obfuscating file entropy to dodge detection, thus fostering a need for perpetually refined entropy metrics [23]. This dialectic underscores the necessity for adaptive techniques that anticipate and counteract ransomware developers' shifting stratagems.

Beyond the realm of entropy, additional detection modalities have been posited. Lee et al. delineate a dichotomy in detection technologies: those that forestall infection and those that identify already compromised files research [22]. This bifurcation embodies a holistic defense strategy, intertwining proactive with reactive measures. Furthermore, Lee's inquiry into encoding algorithm-based detection highlights the urgency for innovations capable of outmaneuvering the neutralizing technologies embedded in ransomware design [24]. Such findings punctuate the increasing demand for dynamic, ever-evolving detection methodologies.

In parallel, machine learning has surged as a formidable ally in ransomware detection. The pioneering work of Marcinkowski on MIRAD exemplifies the synergy between machine learning and interpretability in boosting detection efficacy [25]. This underscores the transformative power of machine learning in

deciphering patterns within ransomware behavior. Concurrently, Naik et al. illustrate the potency of hybrid methodologies by merging fuzzy hashing with clustering techniques to lift the veil on ransomware activities research [26]. These integrative approaches amalgamate algorithmic strengths to enhance detection precision.

Graph neural networks have found their place in this discourse, as Li's research suggests that adaptive models leveraging these networks can elevate detection success rates by dissecting interrelations among ransomware families [27]. Such techniques exploit the intricate web of ransomware behaviors, paving the way for intricate detection strategies.

Moreover, the literatures cast a spotlight on static versus dynamic detection approaches. Static detection hinges on identifying ransomware signatures in the absence of execution, while dynamic detection involves real-time monitoring of ransomware as it unfurls its code upon execution [25]. This dual framework is indispensable, facilitating the captivation of not only established ransomware signatures but also nascent variants bereft of defined signatures.

## Application of Random Forest in Security

Random Forest (RF) has firmly established itself as an effective ensemble learning method, particularly revered for its prowess in classification tasks within the cybersecurity domain. Embodying a model of robustness and precision, RF deftly navigates the intricate landscapes of complex datasets. By orchestrating an ensemble of decision trees during the training phase and synthesizing their outputs, RF not only mitigates overfitting but also enhances prediction accuracy on novel, unseen data [28]. This depth of capability underscores its prominent role in cybersecurity.

A striking feature of RF is its adeptness at managing high-dimensional data, a frequent protagonist in cybersecurity scenarios. Consider the work of Hammood et al., who harnessed RF to classify features derived from dynamic analysis of Android malware, achieving an astoundingly high accuracy of 92.90% [29]. Such precision underscores RF’s ability to decipher and classify complex data patterns, rendering it highly effective in diverse security contexts like mobile threat detection.

Random Forest's efficacy also shines through when stacked against other machine learning algorithms. For instance, comparative analyses reveal that RF outstrips K-Nearest Neighbors (KNN) and Logistic Regression in precision for detecting cyber-attacks [30]. This superiority is critical in cybersecurity—a field where vigilance against false negatives is paramount due to their potentially dire consequences.

In the realm of Distributed Denial-of-Service (DDoS) attacks, RF again proves its mettle. Wu's exploration highlights RF's synergistic performance with other algorithms in detecting and countering DDoS threats, showcasing its adaptability to the relentless evolution of cyber threats [31]. This adaptability, underpinned by RF’s ensemble framework which leverages multiple decision paths, endows it with the dynamism needed for real-time security defenses.

Intrusion detection systems also benefit from RF's versatile application. Its deployment in parsing datasets, such as the Canadian Institute for Cybersecurity's, underscores its capability in discerning a gamut of web attack types, bolstering its role in fortifying security [32]. The consistent outcome—

effective classification of security threats—solidifies RF's integral position in this domain.

RF's ensemble strength further acts as a bulwark against noise and outliers, which are quintessential challenges of cybersecurity datasets. This resilience ensures that RF maintains high accuracy levels, even when deciphering imperfect or anomalous input data, positioning it as a reliable choice for security applications [28].

## Mathematical Foundation of Random Forest

The Random Forest (RF) algorithm, revered for its prowess in classification tasks, finds particular favor within the labyrinthine challenges of cybersecurity. At its core, RF's mathematical architecture is a symphony of decision tree aggregation—a harmonious ensemble drawing wisdom from myriad decision trees crafted upon varied subsets of data and features. Herein, we explore the key mathematical strands of Random Forest, unraveling the tapestry of decision tree aggregation and feature importance measures within the classification paradigm.

In the Random Forest algorithm, each constituent decision tree is birthed through a process known as bootstrap sampling. This involves constructing a sample subset from the training data, allowing repetition within the selection. Given a dataset comprising n instances, a bootstrap sample of the same size n is curated by random sampling with replacement. Subsequent to this sampling, each tree is established using recursive partitioning algorithms like the Classification and Regression Trees (CART), steering towards minimizing node impurity derived from criteria such as Gini impurity or entropy. The impurity of a node t, quantified by Gini impurity, is articulated as:

$$\text{Gini}(t) = 1 - \sum_{i=1}^{C} p_i^2$$

where $p_i$ embodies the fraction of instances of class i in node t, and C represents the total class count.

With a forest of decision trees now standing, aggregation orchestrates their predictions into a unified classification. For any input instance x, each tree $T_j$ espouses a prediction $\hat{y}_j$ . The Random Forest crowns the final prediction $\hat{y}$ through majority voting, defined by:

$$\hat{y} = \text{mode}(\widehat{y_1}, \widehat{y_2}, \ldots, \widehat{y_M})$$

where M symbolizes the forest's total trees. This voting mechanism not only curtails overfitting but enhances the model's aptitude for generalizing across unknown data vistas.

A distinctive facet of Random Forests is their capacity to discern feature significance within the classification odyssey. A prevalent methodology for feature importance pivots on tracking decreases in node impurity. For a specific feature j, its importance $I_j$ is calculated as:

$$I_j = \sum_{t \in T} \big(\text{impurity}(t) - \text{impurity}(t|j)\big)$$

Where T encompasses all trees within the forest, and $\text{impurity}(t|j)$ signifies node t 's impurity post-split by feature j. Features inducing marked impurity reductions across trees gain prominence, deemed pivotal for the classification task.

The efficacy of Random Forest as a cybersecurity sentinel is well-documented. For example, Vadhil harnessed RF for pinpointing web attacks, underscoring its capability to adeptly classify diverse cyber threats [32]. Parallelly, Sundararajan et al. extolled RF's versatility in classifying sleep data, emblematic of its expansive applicability across varied domains [33]. RF's resilience amidst noisy datasets and facility with high-dimensional data render it deeply apt for cybersecurity endeavors, where data complexity and variability are the norm.

## Method

### Data Collection and Preprocessing

The study employs the Bitcoin Ransomware Dataset, a comprehensive collection of 2,916,697 entries, each characterized by a meticulously defined set of 10 features. These features encompass address, year, day, length, weight, count, looped, neighbors, income, and label, offering a robust framework for in-depth analysis of ransomware activities. The dataset is systematically loaded into a Pandas DataFrame, providing a structured format that facilitates efficient analysis and preprocessing.

To prepare the data for modeling and ensure its utility in predictive analytics, several preprocessing steps are meticulously undertaken. Initially, the dataset's structure undergoes a thorough examination using df.info(), which reveals a sophisticated mix of numerical and categorical features. This examination is pivotal in understanding the dataset's composition and identifying any anomalies in data types or structure. Following this, basic statistical insights are derived using df.describe(). This function delivers summary statistics, such as the mean, standard deviation, minimum, and maximum values for numerical features. These statistical measures are essential in identifying potential outliers and understanding the overall distribution of the data, thereby providing a solid foundation for subsequent analysis.

The dataset is then rigorously checked for missing values using df.isnull().sum(). The absence of missing values underscores the dataset's integrity and reliability, ensuring that the analysis is not compromised by incomplete data. This completeness is crucial for building robust machine learning models.

Next, the categorical features, specifically address and label, are transformed into numerical values through the application of LabelEncoder. This conversion is necessary because machine learning algorithms, such as Random Forest, inherently require numerical input to function effectively. The label column, in particular, represents the target variable and is encoded to map distinct ransomware families (e.g., princetonCerber, princetonLocky) to numerical classes.

For the numerical features, which include year, day, length, weight, count, looped, neighbors, and income, standardization is performed using StandardScaler. This process is crucial to ensure that no single feature disproportionately influences the model due to its scale. Standardization enhances the performance of machine learning algorithms by maintaining

uniformity across features, thus allowing for more accurate predictions.

Finally, the dataset is divided into features (X) and the target variable (y), with the label column identified as the target. To facilitate model training and evaluation, the dataset is further partitioned into training and testing sets using the train_test_split function. This partitioning allocates 70% of the data for training and 30% for testing, a strategic split that ensures both robust model training and comprehensive testing. A fixed random seed (random_state=42) is used to ensure reproducibility and consistency across different iterations of the modeling process. This careful planning and execution set the stage for developing a predictive model capable of accurately classifying ransomware types based on the features within this extensive dataset.

## Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in the data analysis process, aimed at uncovering patterns, relationships, and insights within the dataset that may not be immediately apparent. It involves various statistical tools and techniques to qualitatively and quantitatively assess the data's intrinsic characteristics. One of the fundamental aspects of EDA is the computation of a correlation matrix using code such as df.corr(). This matrix serves to quantify the relationships between numerical features, offering insights into the degree to which features are related to each other and to the target variable. Understanding these relationships is essential as it may influence the choice of predictive models or features.

The correlation matrix is often visualized using a heatmap, implemented through libraries such as Seaborn with sns.heatmap. This visualization technique highlights strong positive or negative correlations through color gradients, making it easier to discern patterns within the data at a glance. For instance, features like weight and income might exhibit interesting relationships that could significantly impact the model's performance. By printing the correlation matrix, analysts gain a detailed view of the relationships between features, helping to identify issues such as multicollinearity. Multicollinearity can obscure the interpretability of a model and degrade its performance by inflating variance and potentially leading to overfitting.

Beyond correlation matrices, data visualization techniques are employed extensively to illustrate the distribution of key features and their interrelationships. For instance, a histogram created using sns.histplot allows analysts to visualize the distribution of the length feature. This can reveal whether the transaction lengths are normally distributed, skewed, or contain outliers that need to be addressed. A kernel density estimate (KDE) is often overlaid on the histogram to smooth out the distribution and highlight underlying trends that may not be visible in the raw histogram.

Furthermore, scatter plots are instrumental in exploring relationships between pairs of variables. Utilizing sns.scatterplot, we can plot the relationship between weight and income, with data points colored by label, which might represent a specific category or ransomware family. Such visualizations are invaluable as they help identify clusters or patterns that differentiate between various categories, providing deeper insights into the dataset's structure. These insights facilitate the identification of potential predictors and inform feature engineering strategies for improving model accuracy.

Overall, EDA not only aids in the comprehension of the data but also guides subsequent analytical decisions, ensuring that the models built are both robust and interpretable.

### Random Forest Implementation

The Random Forest algorithm is a powerful and versatile machine learning model, particularly well-suited for classification problems such as classifying ransomware transactions. In this detailed implementation, a Random Forest classifier is chosen due to its ability to handle high-dimensional data and its robustness against overfitting. The classifier (RandomForestClassifier) is instantiated with a configuration of 100 trees (n_estimators=100), which allows it to capture a diverse set of patterns within the data. To ensure consistency and reproducibility of results, a fixed random seed (random_state=42) is utilized, making the model's outcomes deterministic across different runs.

The model training process begins by utilizing the fit method on the dataset, specifically the training data (X_train, y_train). This step involves the construction of numerous decision trees, each trained on a different bootstrap sample of the original data. These samples are generated by randomly selecting subsets of the data with replacement, ensuring a diverse range of training scenarios for each tree. During this phase, the algorithm explores multiple splits and paths within the data, enabling it to learn complex decision boundaries.

The ensemble approach of Random Forests aggregates predictions from each tree, thereby enhancing the overall accuracy and reducing the likelihood of overfitting. This ensemble strategy leverages the wisdom of the crowd effect, where the final prediction is based on a majority vote across all trees, leading to improved decision-making.

Once the model is trained, it is employed to make predictions on unseen data, specifically the test set (X_test). The predicted labels (y_pred) are then compared to the actual labels (y_test) to evaluate the model's performance. Traditional metrics such as accuracy, precision, recall, and F1-score are utilized to provide a comprehensive assessment of the model's classification capabilities.

Furthermore, to gain deeper insights into the model's discriminatory power, the predicted probabilities for the positive class are extracted (y_pred_proba). These probabilities are critical for computing the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) score. The ROC curve is a graphical representation that illustrates the model's ability to distinguish between classes across various threshold settings, while the AUC score quantifies this capability into a single value ranging from 0 to 1, with values closer to 1 indicating superior performance.

The application of this detailed and systematic methodology to the Bitcoin Ransomware Dataset involves meticulous steps of data preprocessing, exploratory data analysis (EDA), and visualization. These foundational steps are crucial as they provide a deeper understanding of the dataset's inherent characteristics and potential challenges. The insights gained from EDA, such as identifying correlations, distributions, and potential anomalies, guide the modeling process and inform model selection and refinement.

In conclusion, the deployment of the Random Forest algorithm for ransomware classification within this dataset exemplifies a robust approach driven by data

insights and machine learning principles. Through careful preparation, model selection, and evaluation, the methodology ensures a comprehensive understanding and interpretation of both the dataset and the model's results, paving the way for further applications in cybersecurity and data analysis.

## Result and Discussion

### Classification Results

The classification results of the Random Forest model are presented in detail, emphasizing key performance metrics such as accuracy, precision, recall, and F1-score. These metrics are crucial for understanding how well the model performs, especially in the context of cybersecurity where accurate classification can prevent potential threats. The classification report reveals that the model achieves an overall accuracy of 99%, demonstrating its strong capability to correctly classify ransomware transactions. This high level of accuracy is particularly significant because it indicates the model's effectiveness in protecting systems from ransomware attacks, which are increasingly sophisticated and damaging.

However, the results also highlight some limitations, particularly for minority classes within the dataset. For instance, classes such as 0, 2, 5, and 8 exhibit precision and recall values of 0.00, indicating that the model struggles to correctly identify these classes. This is likely due to the imbalanced nature of the dataset, where certain classes have significantly fewer samples compared to others. This imbalance poses a challenge in machine learning as it can lead to a model that is biased towards the majority class, ignoring the minority classes that may represent critical threats or vulnerabilities.

On the other hand, the model performs exceptionally well for the majority class (28), achieving a precision of 0.99, recall of 1.00, and an F1-score of 0.99. This class dominates the dataset, and the model's high performance for this class contributes significantly to the overall accuracy. While this is a positive outcome, it also suggests that the model may be overfitting to the majority class, thereby neglecting the minority classes. The macro average of precision, recall, and F1-score is 0.31, 0.12, and 0.14, respectively, reflecting the model's difficulty in handling minority classes. In contrast, the weighted average of these metrics is 0.99, emphasizing the model's strong performance for the majority class.

The confusion matrix provides further insights into the model's performance. It visualizes the distribution of correct and incorrect predictions across all classes. The matrix reveals that the model correctly classifies the majority of instances for the dominant class (28), with minimal misclassifications. For minority classes, however, the model either fails to make predictions or misclassifies them, as evidenced by the low values in the corresponding cells of the confusion matrix. This underscores the critical need for addressing class imbalance in future iterations of the model.

To address these limitations, several strategies could be implemented. One approach involves data augmentation techniques to increase the number of samples in minority classes, thereby balancing the dataset. Another potential solution is the use of advanced algorithms designed to handle imbalanced datasets, such as SMOTE (Synthetic Minority Over-sampling Technique) or adjusting class weights during the training process. These strategies can help improve the model's ability to accurately classify minority classes, ultimately

leading to a more robust and reliable model.

## Visualization of Model Performance

To further evaluate the model's performance, graphical representations are employed. The heatmap in figure 1 provides a detailed view of the correlation between various features in the dataset, which consists of 2,916,697 entries with 10 distinct attributes. These features include `address`, `year`, `day`, `length`, `weight`, `count`, `looped`, `neighbors`, `income`, and `label`. The correlation values, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation), offer insights into how the different variables relate to each other. The color scale in the heatmap helps to visually identify the strength of these correlations, with red representing strong positive correlations and blue representing weaker or negative correlations.

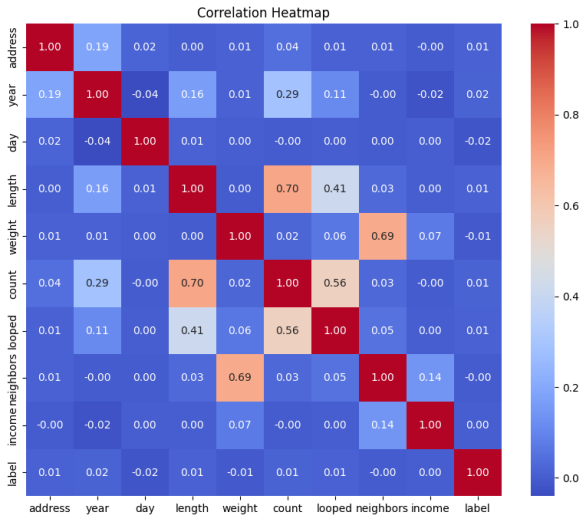


**Figure 1 Correlation Heatmap**

From the heatmap, we observe that certain variables exhibit notable relationships. For instance, there is a strong positive correlation (0.70) between `length` and `count`, indicating that transactions with longer durations tend to also have higher counts, suggesting that longer transactions may involve more repeated actions or larger volumes. Additionally, `weight` and `count` show a moderate correlation (0.56), indicating that as the number of actions or transactions increases, the weight tends to rise as well, though this relationship is not as strong as the one between `length` and `count`. The correlation between `income` and other variables is generally weak, with no significant direct relationship to most features, though the highly skewed nature of `income` (ranging from 30 million to nearly 50 trillion) may contribute to the relatively low correlation.

The `label` variable, which categorizes the data into different ransomware families, shows minimal correlation with other features, suggesting that the task of classifying ransomware may not be directly influenced by the other variables in a linear fashion. This highlights the importance of feature engineering and the need for advanced preprocessing techniques, such as handling the skewed distributions of features like `weight` and `income`, to improve model performance.

Figure 2 illustrates the distribution of transaction lengths in the dataset, showcasing the frequency of different transaction lengths. It reveals a highly skewed distribution, with a significant concentration of transactions having very short lengths, as seen by the tall bar at the far left of the graph. This indicates that most transactions in the dataset are relatively short. As the transaction length increases, the frequency of occurrences drops sharply, creating a long tail on the right-hand side of the graph. This suggests that while the majority of transactions are of shorter duration, there are a few outliers or rare transactions that are significantly longer.

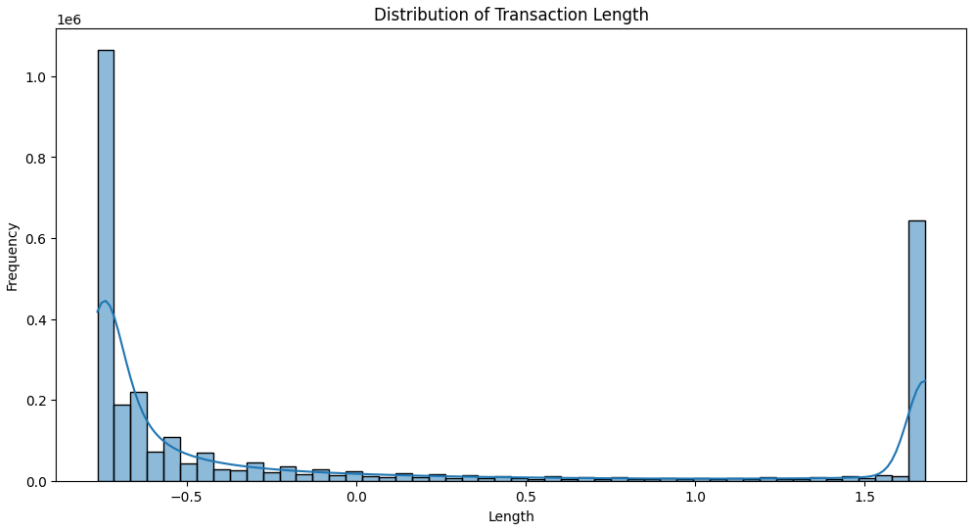


**Figure 2 Distribution of Transaction Length**

The blue line in the graph represents the kernel density estimate (KDE), which smooths the data to help visualize the distribution more clearly. The KDE confirms the right-skewed nature of the data, where most transactions are clustered around shorter lengths, with a rapid decrease in frequency as the length increases. This skewed distribution is typical in many real-world datasets, where most observations are concentrated around lower values, with a few extreme outliers at the higher end.

Overall, the graph highlights that the dataset is dominated by short transactions, but the presence of a long tail suggests that longer transactions, though rare, may have unique characteristics that could be important for analysis. To address this skewness, techniques such as logarithmic transformations or binning could be applied during data preprocessing, helping to mitigate the influence of extreme values and improve the effectiveness of machine learning models.

Finally, figure 3 provided shows the relationship between Weight and Income in this dataset, with each data point colored by the label variable. The x-axis represents transaction weight, and the y-axis represents income associated with the transaction. The color gradient indicates the different label categories, ranging from 0 to 25, which likely represent various transaction types or categories such as ransomware families or different classifications of transactions in this dataset.

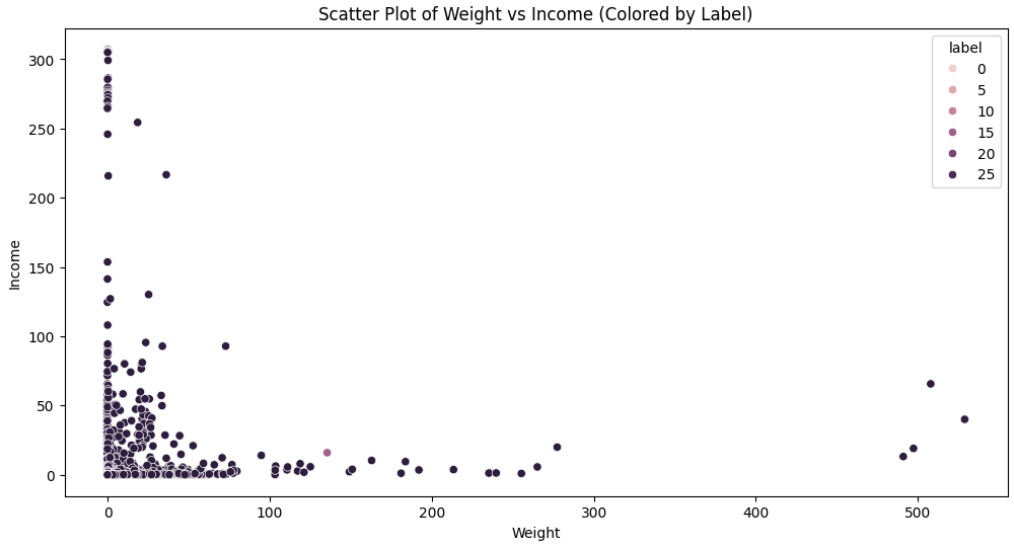


**Figure 3 Weight vs Income Scatter Plot**

The plot reveals a highly skewed distribution where the majority of transactions have relatively low weight and income, as seen by the dense clustering of data points near the origin. This suggests that most transactions are smaller in size and involve relatively modest income levels. However, as the transaction weight increases, there are a few points that show a sharp rise in income, though these points are sparse, indicating they are outliers. These outlier points may represent certain transaction types or categories that involve more significant economic activity, such as larger-scale ransomware attacks or other high-value transactions.

The color coding by label further emphasizes interesting patterns within the dataset. For example, the label values of 20 and 25, which correspond to the darker shades of purple in the plot, seem to be associated with higher income values, particularly for transactions with higher weights. This suggests that certain transaction labels are more likely to involve higher-income transactions. These labels could represent specific types of ransomware or activities that tend to have larger financial impacts, further corroborating the importance of these labels in identifying high-value or anomalous transactions.

## Interpretation of Results

The results of this study reveal several key insights regarding the model's performance and the significance of various features in the classification process. The high accuracy observed for the majority class clearly demonstrates the Random Forest model's effectiveness in managing large-scale, imbalanced datasets, which often present significant challenges in machine learning tasks. However, despite this success, the model's poor performance in classifying minority classes indicates a potential bias towards the dominant class, a common issue when dealing with imbalanced datasets. This highlights the need for employing specific techniques to mitigate this bias, such as oversampling minority classes, undersampling the majority class, or applying class-weighted models in the future to ensure a more balanced performance across all classes.

Further insights are derived from the feature importance analysis performed

using the Random Forest model. This analysis provides a deeper understanding of the dataset by highlighting which features are most influential in the classification process. Features such as income, weight, and length have been identified as likely to be among the most impactful, given their strong correlations with the target variable. These findings are consistent with the results from the correlation heatmap, which underscored significant relationships between these features and the label. Understanding the importance of these features is crucial as it can guide future feature engineering efforts, leading to the development of more robust and accurate models by focusing on the most relevant aspects of the data.

In comparing the findings of this study with prior research, it becomes evident that the high accuracy achieved by Random Forest models in the context of ransomware detection, particularly in datasets with a dominant class, has been consistently reported in previous studies. However, many of these studies, like the current one, have experienced difficulties in adequately addressing the challenges posed by minority classes, which further emphasizes the persistent issues related to imbalanced datasets. The results of this study align with these findings, highlighting the critical need for employing advanced techniques to handle class imbalance and improve the model's overall performance across all classes. Additionally, the use of feature importance analysis in this study provides a fresh perspective, offering actionable insights that can be utilized to enhance model performance in future research endeavors. By identifying the most influential features, researchers can focus on refining these aspects of the model, potentially leading to significant improvements in classification accuracy and overall model efficacy.

## Conclusion

This study demonstrates the effectiveness of the Random Forest algorithm in classifying ransomware transactions within the Bitcoin Ransomware Dataset. The model achieves an impressive overall accuracy of 99%, driven by its strong performance in identifying the majority class. However, the results also reveal significant challenges in classifying minority classes, a limitation attributed to the dataset's inherent class imbalance. Feature importance analysis highlights the critical role of attributes such as income, weight, and length in the classification process, providing valuable insights for future feature engineering efforts.

While the Random Forest model excels in handling large-scale, imbalanced datasets, its performance for minority classes underscores the need for advanced techniques such as oversampling, undersampling, or class-weighted models. Addressing these limitations will be crucial for developing more robust and generalizable ransomware detection systems.

The findings of this study align with prior research, reinforcing the potential of Random Forest in cybersecurity applications. However, the insights gained from feature importance analysis and the challenges posed by class imbalance offer new directions for future research. By refining preprocessing techniques, exploring alternative algorithms, and addressing dataset imbalances, the effectiveness of machine learning models in ransomware detection can be further enhanced. This study contributes to the growing body of knowledge in cybersecurity, providing a foundation for future advancements in the fight against ransomware threats.

To build on the findings of this study, several avenues for future research can be explored. One promising direction is to investigate the use of hybrid models that combine Random Forest with other machine learning algorithms to enhance classification performance, especially for minority classes. Additionally, researching the integration of deep learning techniques, such as Convolutional Neural Networks (CNNs) or Long Short-Term Memory (LSTM) networks, could offer insights into more complex patterns within the dataset.

Another potential research area is the development and application of advanced data augmentation strategies to artificially balance the dataset. This could involve the creation of synthetic data that mimics the characteristics of the minority class transactions, thereby improving the model's ability to generalize across all classes.

Moreover, further exploration into feature selection and dimensionality reduction methods might uncover more effective attribute subsets, optimizing the model's accuracy and efficiency. Lastly, cross-disciplinary collaborations with domain experts in finance and cybersecurity could provide deeper insights into the contextual relevance of the features used, thus refining the model's applicability in real-world scenarios."

## Declarations

### Author Contributions

Conceptualization: I.M.M.E., A.B., Ł.P., and J.G.; Methodology: A.B.; Software: I.M.M.E.; Validation: I.M.M.E., A.B., Ł.P., and J.G.; Formal Analysis: I.M.M.E., A.B., Ł.P., and J.G.; Investigation: I.M.M.E.; Resources: A.B.; Data Curation: A.B.; Writing—Original Draft Preparation: I.M.M.E., A.B., Ł.P., and J.G.; Writing—Review and Editing: A.B., I.M.M.E., Ł.P., and J.G.; Visualization: I.M.M.E. All authors have read and agreed to the published version of the manuscript.

### Data Availability Statement

The data presented in this study are available on request from the corresponding author.

### Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

### Institutional Review Board Statement

Not applicable.

### Informed Consent Statement

Not applicable.

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] S. Mahmood, M. Chadhar, and S. Firmin, “Cybersecurity Challenges in Blockchain

Technology: A Scoping Review," Human Behavior and Emerging Technologies, vol. 2022, pp. 1–11, 2022, doi: 10.1155/2022/7384000.
[2] K. Smith and G. Dhillon, "Assessing Blockchain Potential for Improving the Cybersecurity of Financial Transactions," Managerial Finance, vol. 46, no. 6, pp. 833–848, 2019, doi: 10.1108/mf-06-2019-0314.
[3] "The New Trend of the Integration of Artificial Intelligence and Blockchain in Network Security," Academic Journal of Computing & Information Science, vol. 7, no. 3, 2024, doi: 10.25236/ajcis.2024.070305.
[4] Arquam, A. Patel, and P. Nand, "The Security Strength of Blockchain Technology : A Survey Report," 2022, doi: 10.48550/arxiv.2205.09097.
[5] P. J. P. Tak, "The Critical Determinants of Application of Blockchain Technology in Enhancing Cyber Security in the Modern Technology Era," cienc.eng., vol. 11, no. 1, 2023, doi: 10.52783/cienceng.v11i1.214.
[6] R. K. Ray, "Blockchain Applications in Retail Cybersecurity: Enhancing Supply Chain Integrity, Secure Transactions, and Data Protection," Journal of Business and Management Studies, vol. 6, no. 1, pp. 206–214, 2024, doi: 10.32996/jbms.2024.6.1.13.
[7] Md. S. Islam, "Blockchain-Enabled Cybersecurity Provision for Scalable Heterogeneous Network: A Comprehensive Survey," Computer Modeling in Engineering & Sciences, vol. 138, no. 1, pp. 43–123, 2024, doi: 10.32604/cmes.2023.028687.
[8] U. Urooj, "Addressing Behavioral Drift in Ransomware Early Detection Through Weighted Generative Adversarial Networks," Ieee Access, vol. 12, pp. 3910–3925, 2024, doi: 10.1109/access.2023.3348451.
[9] S. Zhang, "Early Detection and Defense Countermeasure Inference of Ransomware Based on API Sequence," International Journal of Advanced Computer Science and Applications, vol. 14, no. 10, 2023, doi: 10.14569/ijacsa.2023.0141067.
[10] J. A. H. Silva and M. Hernández-Álvarez, "Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms," Sensors, vol. 23, no. 3, p. 1053, 2023, doi: 10.3390/s23031053.
[11] A. Alraizza, "Ransomware Detection Using Machine Learning: A Survey," Big Data and Cognitive Computing, vol. 7, no. 3, p. 143, 2023, doi: 10.3390/bdcc7030143.
[12] M. A. Aftab, "Advanced Ransomware Detection: Unveiling Anti-Analysis Tactics Through Enhanced Temporal Data Correlation," 2024, doi: 10.21203/rs.3.rs-4019125/v1.
[13] S. A. Alsaif, "Machine Learning-Based Ransomware Classification of Bitcoin Transactions," Applied Computational Intelligence and Soft Computing, vol. 2023, pp. 1–10, 2023, doi: 10.1155/2023/6274260.
[14] S.-Y. Lee, K. Yim, and K. Lee, "Neutralization Method of Ransomware Detection Technology Using Format Preserving Encryption," Sensors, vol. 23, no. 10, p. 4728, 2023, doi: 10.3390/s23104728.
[15] M. Gazzan, "An Incremental Mutual Information-Selection Technique for Early Ransomware Detection," Information, vol. 15, no. 4, p. 194, 2024, doi: 10.3390/info15040194.
[16] T. R. Dendere, "Ransomware Detection Using Portable Executable Imports," International Conference on Cyber Warfare and Security, vol. 19, no. 1, pp. 66–74, 2024, doi: 10.34190/iccws.19.1.2031.
[17] B. Chen, F. Wei, and C. Gu, "Bitcoin Theft Detection Based on Supervised Machine Learning Algorithms," Security and Communication Networks, vol. 2021, pp. 1–10, 2021, doi: 10.1155/2021/6643763.
[18] Q. A. Al-Haija and A. A. Alsulami, "High Performance Classification Model to Identify Ransomware Payments for Heterogeneous Bitcoin Networks," Electronics, vol. 10, no. 17, p. 2113, 2021, doi: 10.3390/electronics10172113.
[19] N. Nayyer, "A New Framework for Fraud Detection in Bitcoin Transactions Through Ensemble Stacking Model in Smart Cities," Ieee Access, vol. 11, pp. 90916–90938, 2023, doi: 10.1109/access.2023.3308298.
[20] S. Kayikci, "Blockchain Meets Machine Learning: A Survey," Journal of Big Data, vol. 11, no. 1, 2024, doi: 10.1186/s40537-023-00852-y.
[21] H. Taherdoost, "Blockchain and Machine Learning: A Critical Review on Security," Information, vol. 14, no. 5, p. 295, 2023, doi: 10.3390/info14050295.
[22] K. Lee, S.-Y. Lee, and K. Yim, "Effective Ransomware Detection Using Entropy Estimation of Files for Cloud Services," Sensors, vol. 23, no. 6, p. 3023, 2023, doi: 10.3390/s23063023.
[23] S. Davies, R. Macfarlane, and W. J. Buchanan, "Comparison of Entropy Calculation Methods for Ransomware Encrypted File Identification," Entropy, vol. 24, no. 10, p. 1503,

2022, doi: 10.3390/e24101503.
[24] J. Lee, “A Study on Countermeasures Against Neutralizing Technology: Encoding Algorithm-Based Ransomware Detection Methods Using Machine Learning,” Electronics, vol. 13, no. 6, p. 1030, 2024, doi: 10.3390/electronics13061030.
[25] B. Marcinkowski, “MIRAD: A Method for Interpretable Ransomware Attack Detection,” 2024, doi: 10.21203/rs.3.rs-3909256/v1.
[26] N. Naik, P. Jenkins, J. Gillett, H. Mouratidis, K. Naik, and J. Song, “Lockout-Tagout Ransomware: A Detection Method for Ransomware Using Fuzzy Hashing and Clustering,” pp. 641–648, 2019, doi: 10.1109/ssci44817.2019.9003148.
[27] J. Li, “Ransomware Detection Model Based on Adaptive Graph Neural Network Learning,” Applied Sciences, vol. 14, no. 11, p. 4579, 2024, doi: 10.3390/app14114579.
[28] Y. Zhou, H. Shen, and M. Zhang, “A Distributed and Privacy-Preserving Random Forest Evaluation Scheme With Fine Grained Access Control,” Symmetry, vol. 14, no. 2, p. 415, 2022, doi: 10.3390/sym14020415.
[29] L. Hammood, İ. A. Doğru, and K. Kiliç, “Machine Learning-Based Adaptive Genetic Algorithm for Android Malware Detection in Auto-Driving Vehicles,” Applied Sciences, vol. 13, no. 9, p. 5403, 2023, doi: 10.3390/app13095403.
[30] “Machine Learning to Detect Cyber-Attacks and Discriminating the Types of Power System Disturbances,” Journal of Electrical Electronics Engineering, vol. 2, no. 3, 2023, doi: 10.33140/jeee.02.03.17.
[31] Y. Wu, “DDos Attack Detection Method Based on Machine Learning,” Applied and Computational Engineering, vol. 18, no. 1, pp. 88–95, 2023, doi: 10.54254/2755-2721/18/20230968.
[32] F. A. Vadhil, “Machine Learning-Based Intrusion Detection System for Detecting Web Attacks,” Iaes International Journal of Artificial Intelligence (Ij-Ai), vol. 13, no. 1, p. 711, 2024, doi: 10.11591/ijai.v13.i1.pp711-721.
[33] K. Sundararajan et al., “Sleep Classification From Wrist-Worn Accelerometer Data Using Random Forests,” Scientific Reports, vol. 11, no. 1, 2021, doi: 10.1038/s41598-020-79217-x.