Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 17 (2022) pp. 165-174
© MIR Labs, www.mirlabs.net/jias/index.html

Leveraging Contractive Autoencoder with Fuzzy Lattice Reasoning and Resilient KNN for Detection of multi-level Bitcoin Ransomware

Mrutyunjaya Panda[1] and Ajith Abraham[2]

[1]Department of Computer Science and Applications, Utkal University, Vani Vihar, Bhubaneswar-751004, India
mrutyunjaya74@gmail.com

[2] Machine Intelligence Research Labs, USA,
abraham.ajith@gmail.com

Abstract: **In recent years, ransomware attacks have become increasingly rampant by the offenders for which ransomware has maintained a major cyber security threat as time progresses. With paradigm shift from social to technical factors, ransomware has also maintained the equal adaptiveness by shifting its focus from initial days' scareware and locker attacks to most recent crypto-ransomware threats. There is no silver bullet available to wipe out completely crypto-ransomware attacks for its obvious relationships between social engineering which investigates more infections with encrypted malware. Bitcoin, a means of digital payment demanded by Ransomware family needs characterization and analysis to predict the crypto-ransomware attack types. In this paper, at first, contractive autoencoder (CAE) is used on bitcoin transaction dataset for dimensionality reduction as a filter approach in order to obtain a reduced yet a powerful representation of the raw data and then the output of CAE is applied to the classifier for its improved performance and to make it a robust model. We use two classifiers for our experiments namely: Resilient KNN and Fuzzy Lattice Reasoning (FLR). The original KNN classifier was successful in dealing with homogenous data where the values of the numerical attribute exist completely but poses limitations while dealing with heterogeneous incomplete data containing mixed data (numeric and categorical) yet having missing values. Further, KNN used same K values for all the query objects that sometimes leads to misclassification. Resilient KNN is proposed in this paper to deal with these pitfalls effectively by assigning different k-values for different query objects, so as to obtain a most accurate predictive model. Next, the FLR is used for its ability to handle different types of data types and moreover, it is incremental and fast learning which tempted us to explore its possibility in detecting the crypto-ransomware attacks efficiently. The experimental results with several conventional and new evaluation metrics justifies the suitability of our proposed approach in building a robust and efficient classifier model to detect crypto ransomware families in comparison to existing research.**

Keywords: *bitcoin, Crypto-ransomware, cyber security, autoencoder, FLR, Resilient KNN, Likelihood ratio, Youden's Index, Net benefit, Efficiency index, Gain, detection accuracy*

I. INTRODUCTION

In today's cloud computing world with untraceable payment methods, IT infrastructures becoming a critical one much like roads, electricity and financial infrastructures etc. to organizations. At the same time, cyber criminals also continuously trying to explore several ways and means to attack business infrastructure to gain unlawful interests [1]. With rapid development in software technology, ransomware becomes an armament for the remote attackers. Recently, the so called cybercriminals used crypto-ransomware attacks which seems to be a shift from the early ransomware attacks such as: scareware and lockers ransomware. Through lockers ransomware, the computing system is locked, so that all your files and data are inaccessible to the victim whereas scareware ransomware scares victim into visiting spoofed website which are usually come in the form of pop-up ads or through spam email attacks. Both locker and spareware are ransomware without any encryption, but crypto-ransomware comes with encryption in order to extort money by scrambling the useful data in such a way that it is quite difficult to decrypt by the organization and then, the offenders will seek ransom to be paid for the release of the malware encrypted by themselves [2].

Ransomware is a malware (or malicious software) which seize and obstruct the useful data or steal the computing machine of the victim as late as ransom price is settled up with money. The general life cycle of the ransomware is distributed over six phases [3]: Pahse-1 is distribution of the ransomware which come to the victim as a phishing email containing malicious attachment or a code executioner or through drive by downloading. In Phase-2, ransomware infects the machine by installing itself to pull through a reboot or prohibits shadow copies or current anti-virus processes, while in phase-3, the malware gets in touch with its command and control server for encryption key. In phase-4, the malware scans the victims file in either pdf, word or .png format and then encrypt the selected files using some encryption technique best known to them. Finally, extortion process is carried by sending a note to the victim asking for ransom price. Once this ransom price is received by the attacker, decrypt key is sent to the victim.

With today's digital revolution, rising economy and emergence of rising technologies, creating digital currencies became the most contentious and equivocal in modern digital markets globally as a step towards developing cashless society. People can generate their own money in terms of digital cryptocurrencies. Bitcoin is such a cryptocurrency that is used in the form of digital data where the miners (or users) can send those data by electronic means from their computing device in a peer-to-peer network to make payment if the lender accepts it. Day by day, the number of miners are increasing rapidly to 42 million for 2957 cryptocurrencies and so also the price of the bitcoin. Since the launch of bitcoin in January 2019 to February 2020, the price of a single bitcoin has been increased from $0.0008 to $10,16 [4]. Unlike traditional financial asset series, cryptocurrency's price series cannot be well predicted because of its asymmetric information in financial markets, very little knowledge about them on how it is created as they are not physical currencies and most importantly their chaotic fluctuations due to financial, social and political uncertainties. Hence, accurate prediction and forecasting of bitcoin prices under these uncertain environments may help miners (or users) to have minimum risks and financial losses [4].

With the rising trends of Blockchain as a distributed public ledger which do not need any central authority where the entrenched transaction between two unknown parties recorded in the ledger are publicly available. This led to its first ever application in bitcoin cryptocurrency as Blockchain 1.0 [5] and now, one can see that more than thousands of blockchain based cryptocurrencies are floating in the market [6]. In bitcoin transactions, a payment between two unknown parties can be made by delivering a public bitcoin address as short string to a sender through anonymity networks such as Tor (acronym for Onion routing project where the IP address, online data and browsing history are hidden using a series of layered nodes in the network). This type of bitcoin transactions was also being noticed by the offenders as well which attracts them to get involved in cybercrime activities relating to criminal abuse of blockchain [7]. Recently, it is observed that receiving ransom price through bitcoin become more simple and secure which have never been realized earlier attracted the attention of many [8]. This motivates us to explore further insights in this emerging area to understand its impact on economy as a societal relevance.

The goal of this paper is to investigate properly to develop robust machine learning based solutions to crypto-ransomware detection with several new metrics apart from the tradition performance metrics to justify the effectiveness of the proposed approach. With these approaches, we firmly believe that we could able to address the crypto-ransomware detection algorithms' blind spot and develop a strong defender system.

Therefore, it is of paramount importance for researchers to develop some methodologies for efficient detection and identification of crypto-ransomware attack at an early stage before pre-encryption stage so as to minimize the risk of economic loss.

The rest of the paper is organized as follows. In Section II, literature relating to strength and weakness of ransomware, bitcoin price prediction and attack detection are described to identify the research objectives. Section III presents the details about the dataset used in our research followed by research methodology adopted to undertake this research in Section IV. Section V outlines the experimental setup with results obtained and discuss out findings with a comparison with others work. Finally, we conclude in Section VI with future directions.

II. LITERATURE REVIEW

Corbet et al. [9-10] presented a seminal review paper to discuss about the various useful features of cryptocurrencies. Akshaya et al. [11] discusses about the usefulness of the existing ML and DL techniques to predict the price of cryptocurrencies but there is no comparison about the performance of those methods is presented. In [6], Satoshi Nakamoto has presented blockchain-based digital monetary systems and introduced bitcoin as the first cryptocurrency and made their code available for the public. Ullah et al. [12] used an effective and scalable modified decision tree as online machine learning method to extract good features from ransomware dataset and perform 2-class classification to detect and predict the ransomware efficiently with an extended accuracy of 99.56%. Kok et al. [13], proposed Pre-Encryption Detection Algorithm (PEDA) where secure hashing algorithm was used for signature detection with some known crypto-ransomware signature at first followed by application of learning analytics in the second stage to detect pre-encryption API with cross validation testing. They concluded that PEDA could able to identify 14 APIs in order to differentiate between ransomware and goodware, but the limitation lies is its inefficiency in detection ransomware that uses its own native encryption code. Sebasti ão and Godinho [14] forecasted the profitability and predictability of several machine learning algorithms such as: linear models, Random forest and support vector machine on three major cryptocurrencies including bitcoin, etherium and litecoin. They concluded that the forecast accuracy is different for different models, hence superiority of one above the other may not be possible. Bitcoin price prediction using popular machine learning algorithms such as: Bayesian optimized recurrent neural network (RNN) and a Long Short-Term Memory (LSTM) network along with ARIMA on a bitcoin price index data and found that LSTM outperforms all with an accuracy of 52% [15]. Mudassir et al. [16] proposed machine learning-based bitcoin price prediction models for one, seven, thirty and ninety days with 65% classification accuracy for next-day and 62% to 64% accuracy for seventh–ninetieth-day forecast respectively. Beaman et al [17] highlighted some recent advances in ransomware analysis, detection and prevention techniques with a view that there is a trend in using machine learning based approaches to detect ransomware with research limitations and future directions. B. Irigoyen et al. [18] focused on detection techniques with the core focus on crypto ransomware and proposed some criteria including Blockchain technology, machine learning etc. to detect and prevent ransomware threats in health care systems. Yaqoob et al. [19] proposes some case studies to aware the users and manufacturers about the vulnerabilities of IoT devices along with open research challenges while developing a smart and secured system. In Aurangzeb et al. [20], the authors conducted an extensive review on emerging ransomware attacks and suggested few security challenges to counter such a situation. Al-Haiza and Al Sulami [21] proposes supervised machine learning methods such as: shallow neural networks (SNN) and optimizable decision trees (ODT) for classification of crypto-ransomware payment patterns, and finally, concludes that model based decision trees (ODT) outperforms the SNN with predictive accuracy of 99.9% and 99.4% ransomware payments in bitcoin transaction for two-class and multiclass classification respectively. Kok et al. [22] uses Random Forest learning Analytics algorithm on Ransomware detection in three different datasets such as: pre-encryption (PE), VirusTotal and theZoo. They used conventional and new performance metrics and concluded that the new metrics are capable of correct prediction of ransomware with a highest net benefit of 0.7817 for PE dataset in comparison to others.

Different from related studies, the contributions in this paper are as follows.

C1. Application of novel unsupervised deep learning-based preprocessing methods such as contractive autoencoder (CAN) to produce a compressed representation raw data suitable for classification tasks.

C2. Two classification models with Resilient KNN and Fuzzy lattice reasoning are explored to build a strong defender system against the crypto-ransomware attacks.

C3. Seven new performance metrics such as: likelihood ratio (LR), Diagnostic odd ratio (DOR), Youden's Index (Y), Number needed to diagnose (NND), number needed to misdiagnose (NNM), net benefit (NB) and Efficiency Index (EI) are explored apart from conventional ones to remove the detection algorithms black holes if any.

C4. Finally, comparison with other existing machine learning models are performed and understand the robustness of our approach.

III. CRYPTO RANSOMWARE DATASET USED

The proposed algorithms are applied on the dataset that was created based on extracted daily transaction network to obtain an entire Bitcoin transaction graph from 2009 January to 2018 December using a time interval of 24 hours. In this data creation, network edges that transfer less than B0.3 thresholds are being removed, as hardly any ransom amounts are found below this threshold. Heterogeneous Network address for Ransomware are taken from three widely adopted studies: Montreal, Princeton and Padua [23] in a 24-hour snapshot basis where each address contains the six features such as: income, neighbors, weight, length, count and loop. It is to be noted here that a total of 61,004 addresses were selected from 24 ransomware families where at least one address must appear in more than one 24-hour time window. In this dataset, 13 addresses of the CryptoLocker ransomware appears more than 100 times each and its address "1LXrSb67EaH1LGc6d6kWHq8rgv4ZBQAcpU" appears for a maximum of 420 times. Further, four network addresses of Montreal and Padua ransomware contains conflicting ransomware labels between them. There are one and two P2SH addresses starting with '3' also available in case of APT (Montreal) and Jigsaw (Padua) ransomware respectively. The rest of the heterogeneous network addresses are ordinary addresses staring with '1'. More details about this dataset is presented in Table 1.

The features in the bitcoin transaction graph are specifically designed to understand the bitcoin transaction patterns. Loop feature is used to find the number of bit coin transactions are used in splitting the coin, moving of those coins in different network paths and finally merging them in a single address. It is this final address from where the bitcoins can be sold out and then may be converted to the fiat currency.

In bitcoin transaction dataset, number of transactions with output addresses are less than that of the input ones. While details about the number of bitcoin transactions are obtained from the attribute 'count', the percentage of these transactions' output as amount of information are obtained from the attribute 'weight' and number of mixing rounds on bitcoin transactions are obtained from attribute 'length'. In length attribute, multiple round of transmission of same amount of bitcoin transactions are created with new addresses by hiding the origin of the bitcoin. Out of total 800K bitcoin addresses daily, white bitcoin addresses are capped at 1K per day. At the same time, it is also not clear that whether all white addresses are goodware or ransomware even though the labels of all ransomware families are well known. Ransomware families has right skewness in data distribution in comparison to the goodware (i.e. white) ones.

IV. RESEARCH METHODOLOGIES

In this section, methods adopted for experiments are discussed with their suitability in this research.

A. *Contractive Autoencoder (CAN)*

Autoencoder is a type of neural network structure which is popularly used in machine learning for dimensionality reduction and to uncover details from the compressed raw data. It is comprising of an encoder which compresses the input data and then a decoder sub-models recreates the input from the compressed input data from the encoder stage. Once the model is trained, the encoder model is saved for feature extraction and subsequent application of machine learning techniques while discarding the decoder. For this, auto encoder model is considered as self-supervised model, an unsupervised model using supervised techniques [24]. Autoencoder are type of model that are utilized in several methods to improve performance by manipulating hidden layers.

The prime goal of basic autoencoder is to have dimensionality reduction or noise reduction in the input dataset as a pre-processing step and then reorganize the reduced data as close as possible to the input ones which is shown in Figure 1.

From Figure 1, one can see that important input characteristics of the data are obtained in encoder stage and then, decoder presents outputs similar to input ones by removing the input noise.

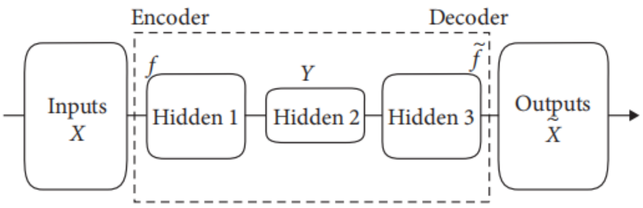

Figure 1. Basic autoencoder architecture

Here, $Y = f(w1.X + \text{Bias})$ and $\tilde{X} = \tilde{f}\,(w2.Y+\text{Bias})$, where w1 and w2 are the hidden layer representation at input and output; f performs compression of the input data into latent space whereas $\tilde{f}$ reconstructs the input back from it.

Table.1 Description of the Bitcoin Ransomware Attack Dataset

No.	Attribute Name	Attribute type	Attribute description
1	Address	String	Bitcoin address specifying the type of transactions in this dataset, Attack (Ransomware) or normal (white or goodware)
2	Day	numeric	The day in the Year of transaction, 1 represents 1st day and 365 as last day of the year
3	Year	numeric	The transaction year
4	Length	numeric	The total number of nonstarter transactions on its longest chain.
5	Weight	numeric	The sum of a fraction of coins that are constructed from the initial transaction and end up at the address.
6	Count	numeric	The number of first-time transactions Associated with an address during a chain of events.
7	Looped	numeric	The number of starting transactions associated with an address that has more than one immediate arc is shown.
8	Neighbors	numeric	Some transactions contain the address as an output, whereas others do not.
9	Income	numeric	The total amount of outputs from the coin to the destination address. Satoshi amount (1 bitcoin = 100 million satoshis).
10	Class/ Label	categorical	Transaction pertains are listed below. It is either a white or non-Ransomware category, which indicates that the transaction is risk-free or one of the 24 Ransomware Categories (e.g., Cryptxxx, CryptoLocker etc.), which indicates that the transaction is risky

For linear autoencoder, having linear activation functions and number of hidden units are less in comparison input dimensions, it creates bottleneck where the encoder learning parameters become the subspace of the principal components of the input space. This under complete autoencoder do not need any regularization as they maximize the probability of the data rather than copying the input to output which learn and describe the latent attributes of the input data. The drawbacks of undercomplete autoencoder lies in misunderstanding of the important features, too lossy, less training data and imperfect decoding etc. However, with the use of nonlinear activation functions in autoencoder, more useful features larger than the input dimensions are obtained in comparison to the features that could have been obtained using conventional principal component analysis. This over complete settings of non-linear auto encoder needs some kind of regularization strategies to abstain from uninteresting findings from such a setup. It is pointed out that nonlinear over complete autoencoder models are more robust in presence of noise and greater flexibility in learning the best features from the training data [25].

A contractive autoencoder (CAN) is a robust autoencoder [26] which is less sensitive to the small variations in the training data by adding some regularization method such as: Frobenius norm of the Jacobian matrix of the encoder activations with respect to the input and then minimize the regularizer. Frobenius norm of Jacobian matrix of the hidden layer is calculated as sum of squares of all input elements. This is represented in equation (1).

$$\alpha(h)=\beta\left\|\frac{\partial f(wX+bias)}{\partial X}\right\|_f^2 \qquad (1)$$

Where α(h) is the Frobenius norm of Jacobian matrix of the hidden layer and β is a free parameter.

B. *Resilient K-Nearest Neighbor (Resilient KNN)*

The basic principle of K-nearest Neighbor (or K-NN) classifier finds the data having k-shortest distance nearest to the training data as their K-nearest neighbors. This is a nonparametric classifier where no grouping is done based on the data distribution which is simple and easy to implement. This method is efficient in noisy data and when training data is huge. However, the weaknesses lie in chosing the optimal value of K nearest neighbors, best distance metric to be used, best attribute selection and more importantly its distance calculation for every instance to the whole training data makes it computationally expensive. Further, it is mainly found suitable for applications with homogenous dataset having complete numerical features only. However, it could not deal with heterogeneous incomplete data with mixed attributes (numeric and categorical) with missing values [27]. Even though there are several modifications to original K-NN by either using imputation or elimination, still it does not prove to be efficient in accurate classification. Hence, we propose to apply a rough set based resilient K-NN (or Rseslib KNN) [28] as an alternative to this, by implementing a fast neighbor search strategy to develop an efficient classifier for very large dataset where we could effectively address incompleteness, heterogeneity and optimal k-value with improved accuracy in classification. The pseudo-code of proposed Resilient KNN is presented in Algorithm 1. More details can be found in [27].

Algorithm 1. Pseudo-code for Resilient KNN

```
Input: Bitcoin Ransomware dataset and feature vector of query object
Output: Ransomware class of query object
Step-1: Find all set of categorical neighbors. Two objects in the dataset are categorical neighbors if they have a common value for at least one categorical feature while others might be similar in some other direction
Step-2: Categorical feature bond is computed by calculating the degree of similarity between the two values of some feature in the query object and a categorical neighbor object in the training dataset, with following sample probability estimates:
    A) For a feature in training dataset, if the query object and neighboring categorical object perfectly matches with no missing values for the considered feature, then the estimated probability is considered to be 1.
    B) If out of two values from the considered feature, one mismatches and the other matches, then the probability estimate is 0.5
    C) If both mismatches are found while comparing, then the probability estimate is 0.25
Step-3: Obtain the categorical object bond by summing all categorical feature bond as per Step-2 for all features in the training dataset
Step-4: Calculate cardinality of the set (J) of nearest categorical neighbors of query object and then calculate the distances between each neighboring object with the query object, and sorted those distances in descending order. We use Euclidean distance for this calculation.
Step-5: For $\lambda$ being the $J^{th}$ of those sorted distances, the set of nearest numerical neighbors are obtained from the set of objects whose distances to the query object are either equal to less than $\lambda$.
Step-6: Find out the true neighbor. An object in the training dataset is said to be a true neighbor of the query object, if it is both a nearest categorical neighbor and a nearest numerical neighbor, to query object.
Step-7: Finally, the class label of query object is determined by using majority voting of true neighbors, where the true neighbors to vote in order to predict the class label of the query object.
```

C. *Fuzzy Lattice Reasoning (FLR)*

The conventional Fuzzy Inference system (FIS) is a fuzzy logic based granular rule induction and generalization method, which when added with neural network has proved to be good at parallel implementation [29]. Fuzzy inference system using lattice theory is introduced much later to effectively classify the data in disparate domain. Then, the concept of Fuzzy lattice reasoning emerged.

A lattice is a partially ordered set in which any two elements have both a greatest lower bound and a least upper bound. Lattice theory emerges naturally in granular computing because (information) granules are partially ordered.

The fuzzy lattice framework is a rule based, reasoning method which is considered to be an artificial neural network based machine learning approach for classification problems. In one way, fuzzy lattice reasoning (FLR) considers fuzzy lattice elements as antecedents and fuzzy inclusion measures as consequences in a fuzzy rule representation. In other way, FLR is proposed to use positive value function such as: linear or non-linear activation functions, as fuzzy lattices from a partially ordered sets. Here, we use non-linear sigmoid function for our experiments for improved performance of the classifier in detecting the attacks.

Fuzzy Lattice Reasoning (FLR) [30] is a classifier which extracts rules from the input data based on fuzzy lattices. The importance of chosing this classifier lies in its ability handle the variety of data types including symbols, graphs, images, fuzzy sets, real vectors and their combinations. Apart from dealing with both point and interval type of data, it can also handle both complete and incomplete lattices. Further, stable yet incremental and fast learning makes FLR is suitable in many machine learning applications [31].

The rules generated by FLR is not known Apriori rather it is incremental rules on the go while training, so if at all more training data is added in this online process, previous data is not lost, helps the model to be improved with new rules [30].

A fuzzy lattice rule is a pair <x,y> where x is an element in fuzzy lattice <L, µ> where L is the complete lattice with membership function µ such that $LxL \rightarrow (0,1)$ and $\mu(x,y)=1$ with $x \leq y$ [32]

A fuzzy relation between two objects may preserve the fuzzy relation between the corresponding two objects in the original set after transformation by using positive evaluation function. The positive valuation function may be treated as an order preserving function as follows: For two lattices L and M, a valuation function is the order preserving mapping (Q) with $L \rightarrow M$, $\forall x, y \in L$, $Q(x) + Q(y) = Q(xUy) + Q(x \cap y)$ and in case where Q(.) satisfies $x< y \Leftrightarrow Q(x) < Q(y)$, the valuation function is termed as positive [31].

Further, it is to be noted here that Fuzzy Lattice is quite different from L-fuzzy set and type-2 fuzzy set where in the first case, mapping is from universe of discourse to a mathematical lattice representation and in the other, mapping to the collection of either conventional fuzzy sets or of fuzzy intervals are used [33]. A pseudo code for training FLR is presented in Algorithm 2.

Algorithm 2. Pseudo-code of Fuzzy Lattice Reasoning (FLR)

```
Step-1: Input dataset to the FLR classifier in terms of $(d_i, O_k)$ where $O_k$ is the output class label of the data $d_i$ with a rule based induction as if data is $d_i$ then class label is $O_k$. Accordingly, a knowledge base is created with all collections of $(d_i, O_k)$ from the dataset.
Step-2: Initially, input $(d_0,O_0)$ and all class labels $C_i$ are memorized by the classifier with initial class label $C_0$.
Step- 3: Now, using inclusion property, next input $(d_i, O_k)$ with $i=1,2..,p$ is compared with already stored set of rules in knowledge base.
Step-4: If it matches with initial rule set as in Step-3, then competition occurs among the "set" rules and classifier returns the winner rule $(d_j, O_j)$ where $j= argmax[i(d_0, d_i], i=1,2,...o$.
Step-5: Now, using assimilation condition: if winner rule $d_j$ and initial input data $d_0$ are having same class label $O_j$ and if size $(d_0 U d_j)$ is less than a user specified threshold, then replace $d_j$ by $d_0 U d_j$ and update the winner rule in the knowledge base. The user threshold can be obtained by FLR with non-linear valuation function used.
Step-6: else, the process is repeated till no more rules are left
```

V. Experimental Setup and Findings

In this research, we propose to address some of the future research directions mentioned by Urooj et al. [34] in terms of getting a robust classifier with (i)a rich dataset, (ii) feature reduction using deep learning, (iii) using fuzzy logic modelling and (iv) computationally efficient. Here, we use a hybrid model by combining contractive autoencoder with Resilient KNN and Fuzzy lattice reasoning to detect ransomware families efficiently. The experimental setup is shown in Figure 2. All the experiments are conducted in this research uses an Intel Core CPU with 8 GB RAM, 1TB HDD in a 64-bit Windows 10 operating system with Java and R as programming environment.

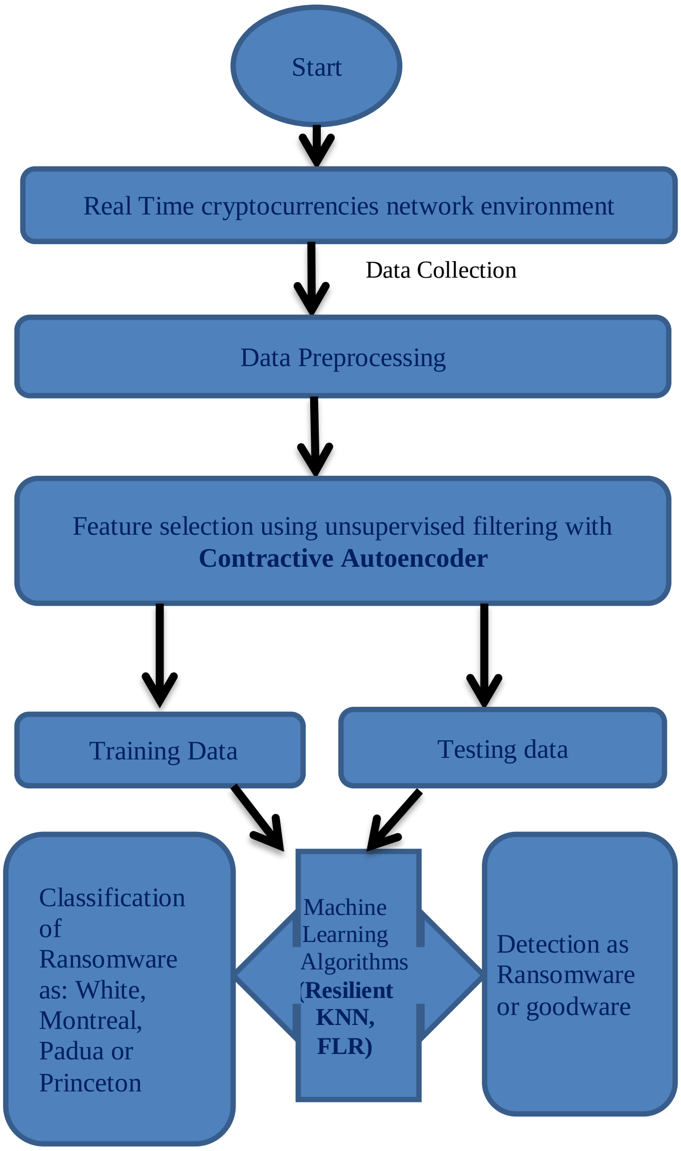
Start
Real Time cryptocurrencies network environment
Data Collection
Data Preprocessing
Feature selection using unsupervised filtering with Contractive Autoencoder
Training Data
Testing data
Classification of Ransomware as: White, Montreal, Padua or Princeton
Machine Learning Algorithms (Resilient KNN, FLR)
Detection as Ransomware or goodware

Figure 2. Proposed Experimental Framework

From Figure 2, it is evident that our experiment on bitcoin transaction recognition model where the prime objective is to develop a self-resilient machine learning and deep learning based classification model that can investigate on the prescribed features of the heterogeneous bitcoin transaction to identify whether the transaction belongs to either a normal or ransomware payment, for a safe cryptocurrency transaction. The experimental setup consists of five stages. At first data is collected followed by data preprocessing using unsupervised instance based selection using remove fold with number of fold=10 on total dataset having 2,91,670 instances. Autoencoder either MLP Autoencoder or contractive autoencoder is used for Feature selection in 3rd stage to obtain the best features for machine learning applications. At fourth stage, the filtered data is separated with 66% training and the rest is for testing of the model which were applied then in the fifth stage with several machine learning algorithms such as: Resilient KNN and Fuzzy lattice reasoning (FLR) to efficient detection of ransomware. Finally, the effectiveness of the model is verified with several performance evaluation metrics both conventional (Accuracy, True positive rate, False positive rate, Precision, Recall, F-measure, training and testing time) and new ones [22] including log likelihood ratio (LR), Diagnostic odd ratio (DOR), Youden's Index (Y), Number needed to diagnose (NND), number needed to mis-diagnose (NNM), Efficiency Index (EI) and Net benefit (NB). The details of the evaluation metrics are provided in Table 2.

A. Results and discussions:

The experimental results and discussions are presented in this section. The comparison with existing research using the conventional metrics for classifier evaluation is presented in Table 3.

The conventional metric has several weaknesses. First, they do not provide enough details about the differentiation between positive and negative outcomes, which is most important in ransomware detection. They also fail to point out the range at which optimal classification may fall for success and failure. Finally, they do not indicate about the net benefit in a chosen classification model. Hence, new evaluation metrics were proposed along with the conventional ones for better detection and prediction of existing and new ransomware families.

Likelihood ratio (LR): The likelihood of identification of API pattern whether belongs to a ransomware or not is measured through likelihood ratio (LR) [45]. LR can be classified as PLR (positive LR) or NLR (negative LR). For strong differentiation between ransomware and white, it is recommended that PLR should be greater than 10 and NLR should be less than 0.1.

Diagnostic odds ratio (DOR): The ratio between PLR and NLR determines DOR, which ranges from zero to infinity. Generally, if DOR is more than 100, then it is assumed that it can differentiate between ransomware and white in a strong way [46].

Youden's index (Y): Youden's index (Y) is used to check whether model developed is predictive correctly or not. For Y=1, a perfect prediction by classifier model with no false positives or false negatives [46].

Number needed to diagnose (NND): This enables us to understand for a single correct positive prediction, how many number of data points are required [46]. The smaller NND value is considered as providing better prediction by the classifier model.

Table 2: Performance Metrics for evaluation of classifiers

Conventional Metric	Definition
TPR (true positive rate)	True Positive/(True Positive + False negative)
FPR	False positive/(False positive + True Negative)
Precision	True Positive/(True Positive + False positive)
Recall	True Positive/(True Positive + False Negative)
Accuracy	(TP + TN)/(TP + FP + FN + TN)
F-score or dice coefficient	2*Precision*Recall/(Precision + Recall)
PLR	TPR/(1-TNR)
NLR	(TPR-1)/TNR
DOR	PLR/NLR
NND	$1/[TPR-(1-TNR)]^{1/Y}$, for Y=1, NND=1/Y
NNM	NNM = 1/Inaccuracy, where Inaccuracy = (FP + FN)/(TP + FP + FN + TN)
Y	TPR-TNR-1
NB	(TP/n)-[(FP/n)*(p/1-p)], n is total number of data and P is the probability threshold ranging from 10% to 99%
Efficiency Index (EI)	(TP+TN)/ (FP+FN), the value of EI lies between 0 to infinity for inaccurate and perfect test.

Table 3: Comparison with traditional performance metrics

Algorithm	Class	TPR	FPR	Precision	Recall	Accuracy in %	F-Score	Training time in second	Testing time in second
Decision Table[35]	multi	0.93	0.012	0.924	0.93	92.97	0.925	103	-
PART [35]	multi	0.96	0.007	0.959	0.96	96.01	0.956	1609	-
SNN+ODT [36]	multi	-	-	0.994	0.993	99.49	99.35	-	-
LSTM [37]	2	-	-	-	-	98	-	-	-
J48 [38]	2	-	-	-	-	97.1	-	-	-
CNN [39]	2	-	-	-	-	97.1	-	-	-
RF [40]	multi	-	-	-	-	84	-	-	-
RF [41]	multi	-	-	-	-	95.7	-	-	-
LSTM [42]	multi	0.97	0.027	-	-	-	-	-	-
SVM [43]	multi	-	-	-	-	-	-	-	-
GTB [44]	multi	-	-	-	-	-	-	-	-
Resilient KNN (ours), k=99 optimal	**multi**	**0.986**	**0.898**	**1**	**1**	**-**	**1**	**150.46**	**10.34**
Contractive Autoencoder+ FLR (proposed)	**multi**	**1**	**0**	**1**	**1**	**88.7**	**1**	**135.24**	**3.98**
Contractive Autoencoder +Resilient KNN, k=10 optimal (proposed)	**multi**	**1**	**0**	**1**	**1**	**97.3**	**1**	**346.69**	**5.52**

Table 4: Comparison with related works using new metric

Algorithm/ Metric	**PLR**	**NLR**	**DOR**	**Y**	**NND**	**NNM**	**NB 10%**	**NB 50%**	**NB 99%**	**EI**
TDA [23]	0.21	-	-	-	-	-	-	-	-	-
DBSCAN [23]	0.04	-	-	-	-	-	-	-	-	-
XGBoost [23]	0	-	-	-	-	-	-	-	-	-
Random Forest [23]	0	-	-	-	-	-	-	-	-	-
Resilient KNN (ours)	**1.09**	**0.14**	**7.785**	**0.086**	**11.627**	**52.63**	**0.108**	**0.973**	**96.327**	**49.44**
Fuzzy Logistic Reasoning (FLR) with Contractive autoencoder (proposed)	∞	**0**	∞	**1**	**1**	**37.03**	**0.032**	**0.291**	**28.866**	**33,052**
Resilient KNN with Contractive autoencoder (proposed)	∞	**0**	∞	**1**	**1**	**8.849**	**0.032**	**0.291**	**28.866**	**7,627**

Number needed to misdiagnose (NNM): This is just the case opposite to NND where we need to find out the number of data points required for making incorrect predictions by the model. For better prediction performance, higher NNM value is recommended [46].

Net benefit (NB): The Net benefit provides detail about the correct or misclassification based on a cutoff point or threshold probability for different exchange rates ranging from 10% to 99% [47].

A comparison with existing work using new metrics are presented in Table 4.

The experimental results show that the ransomware from all sources is infinite I.e. having a PLR greater than 100, and an NLR=0, which indicates that the classifier model has good positive and negative likelihood values. Further, DOR with infinite value indicates that the proposed models can very well distinguish between ransomware and goodware. It can also be seen that both our proposed models have Y=1, means that no incorrect classification is made by the chosen classifiers. Finally, NND of 1 and NNM with 37.03 and 8.849 for Resilient KNN and FLR respectively, suggests the effectiveness of both the models in correct classifications of ransomware families and goodware. While comparing with other works, it can be observed that our both models resilient KNN and FLR with contractive autoencoder outperforms TDA [23] and DBSCAN [23] with infinite PLR value, making them the winner in terms of strong prediction.

From results presented in Table 4, we can see that both of our proposed models Contractive autoencoder with FLR and Resilient KNN produces best prediction in terms of highest PLR, zero NLR, infinite DOR, Y=1, NND=1 with net benefit (NB) for three different ranges 10%, 50% and 90% with 0.032, 0.291 and 28.866 respectively. However, large NNM value with 37.03 for contractive autoencoder with Resilient KNN is the best predictive performance in comparison to its FLR counterpart having NNM=8.849. In comparison with [24], it can be seen that Random forest and XGBoost are having PLR=0 indicating very poor prediction with little poor prediction in case of TDA and DBSCAN with PLR=0.21 and 0.04 respectively.

From Table 4, it is also observed that Efficiency Index (EI) value for FLR with contractive autoencoder is highest with 33052 in comparison to others, making it a most efficient model.

VI. Conclusions and Future directions

In this research work, application of contractive autoencoder as an attribute selection technique is explored to build a robust and efficient prediction model. We have implemented Fuzzy lattice reasoning (FLR) and Resilient K-NN prediction model to perform multi-class ransomware classification as well as 2-class classification (ransomware or white). Separate training and testing dataset were taken to evaluate the performance of the implemented model to its effectiveness in correct prediction. The proposed Contractive autoencoder with Frobenius norm of the Jacobian matrix with sigmoid function performs better than the regular and de-noising autoencoders in terms of robustness and computational cost. The output of the contractive autoencoder is then applied to classifiers such as: Resilient K-NN and FLR for prediction and found that both the models present 97.3% and 88.7% classification accuracy respectively. We also evaluated the experimental results with other conventional metrics to evaluate the proposed models in comparison to others existing research and found interesting results. However, in order to evaluate the classifier performance better, seven new evaluation metrics are proposed in this research that includes: likelihood ratio, diagnostic odds ratio, Youden's index, number needed to diagnose and number needed to misdiagnose, net benefit and efficiency index. From these new metrics comparisons, we could able to conclude that resilient KNN and FLR both with contractive autoencoder perform equally well but best in comparison to TDA, DBSCAN, Random Forest and XGBoost trees with high PLR. However, high NNM value for Resilient KNN makes it a better model in comparison to FLR with less number of incorrect predictions. But at the same time, from efficient index comparison, it is imperative to know that FLR is more efficient than resilient KNN with a very large value.

As a future research direction, we aim at developing new methodologies and new datasets for detection and prediction crypto-ransomware which can help the researchers, practitioners and society at large in decision making process.

References

[1] U. Franke, "The cyber insurance market in Sweden". Comput. Secur. Vol. 68, pp. 130–144, 2017.

[2] Lena Y. Connolly, David S. Wall, "The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures", Computers & Security, Vol. 87, ID- 101568, 2019.

[3] McAfee. "Understanding Ransomware and Strategies to Defeat it", 2018.

[4] H. Pabuçcu, S. Ongan and A. Ongan, "Forecasting the movements of Bitcoin prices: an application of machine learning algorithms", Quantitative finance and economics Journal, Vol. 4, No. 4, pp. 679–692. 2020.

[5] M. Swan, "Blockchain: Blueprint for a new economy". O'Reilly Media, Inc., 2015.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[7] C. Ramey, "The crypto crime wave is here," Wall Street Journal, April 28, 2018

[8] J. Hernandez-Castro, E. Boiten, and M. Barnoux, "The 2nd kent cyber security survey," Kent University reports, 2014.

[9] S. Corbet, B. Lucey, A. Urquhart and L. Yarovaya, "Cryptocurrencies as a financial asset: A systematic analysis," International Review of Financial Analysis, Elsevier, vol. 62(C), pp. 182-199, 2019.

[10] N. Kyriazis, S. Papadamou, and S. Corbet, "A systematic review of the bubble dynamics of cryptocurrency prices," Research in International Business and Finance, Elsevier, vol. 54(C), 2020.

[11] R. Akshaya, B. Eswari, S. Dharani and R. Lalitha, "A survey on anticipation the prices of crypto currency using deep learning: International Journal for Research in Applied Science & Engineering Technology, Vol. 7(3), pp. 1639–1644, 2019.

[12] F. Ullah, Q. Javaid, A. Salam, M. Ahmad, N. Sarwar, D. Shah, and M. Abrar, "Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls", Scientific Programming, Vol. 2020, Article ID 8845833, 10 pages., 2020.

[13] S.H. Kok, Azween Abdullah, NZ Jhanjhi, "Early detection of crypto-ransomware using pre-encryption detection algorithm", Journal of King Saud University – Computer and Information Sciences, 2020.

[14] H. Sebastião and P. Godinho, "Forecasting and trading cryptocurrencies with machine learning under changing market Conditions", Financ Innov, Vol. 7(3), pp. 1-30, 2021 Springer.

[15] S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning", In: 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 339-343, 2018, IEEE.

[16] M. Mudassir, S. Bennbaia, D. Unal and M. Hammoudeh, "Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach", Neural Computing and Applications, 2020.

[17] C. Beaman, A. Barkworth, T. David Akande, S. Hakak and Md. Khurram Khan, "Ransomware: Recent advances, analysis, challenges and future research directions", Computers and Security, Vol. 111, ID-102490, 2021.

[18] W. I. Berrueta, O. D. Morató, L. E. Magaña , A. M. Izal, "A survey on detection techniques for cryptographic ransomware". IEEE Access, Vol. 7, pp. 144925–144944, 2019.

[19] I. Yaqoob, E. Ahmed, M. ur Rehman, A. Ahmed, M. Al-garadi, M. Imran and M. Guizani, "The rise of ransomware and emerging security challenges in the internet of things". Comput. Networks, Vol. 129, pp. 444–58, 2017.

[20] S. Aurangzeb, M. Aleem, M. Iqbal, M. Islam M et al. "Ransomware: a survey and trends", J. Inf. Assur. Secur, Vol. 6(2), pp. 48–58, 2017.

[21] Q. A. Al-Haija, A. A. Alsulami, "High Performance Classification Model to Identify Ransomware Payments for Heterogeneous Bitcoin Networks". Electronics, Vol. 10, no. 2113, 2021.

[22] S. H. Kok, A. Azween and N.Z. Jhanjhi, "Evaluation metric for crypto-ransomware detection using machine learning", Journal of Information Security and Applications Vol. 55, ID-102646, 2020

[23] C.G. Akcora, Y. Li, Y. R. Gel, and M. Kantarcioglu, "BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain". In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Special Track on AI in FinTech., pp. 4439-4445.

[24] Y. He, A. Carass, L. Zuo, B. E. Dewey, J. L. Prince, "Autoencoder based self-supervised test-time adaptation for medical image analysis", Medical Image Analysis, Vol.72, 102136, 2021, Elsevier.

[25] Q. Kong, A. Chiang, A. C. Aguiar, M. Giselle Fernández-Godino, S. C. Myers, D. D. Lucas, "Deep convolutional autoencoders as generic feature extractors in seismological applications", Artificial Intelligence in Geosciences, Vol. 2, pp. 96-106, 2021.

[26] S. Rifai and P. Vincent, X. Muller, X. Glorot and Y. Bengio, "Contractive Autoencoders: Explicit Invariance During Feature Extraction", In: ICML'11: Proceedings of the 28th International Conference on International Conference on Machine Learning, June 2011, pp. 833–840.

[27] A. Hamed, M. Tahoun and H. Nassar, "KNNHI: Resilient KNN algorithm for heterogeneous incomplete data

classification and K identification using rough set theory”. Journal of Information Science. pp. 1–25, 2022, Sage Publication.

[28] A. Wojna and R. Latkowski. “Rseslib 3: Library of Rough set and machine learning methods with extensible architecture”, IN: J. Peters and A. Skowron edns., Transactions on Rough Sets XXI, LNCS, Vol. 10810, Springer, 2019.

[29] S. Paul and S. Kumar, “Subset hood-product fuzzy neural inference system (SuPFuNIS)”, IEEE Transactions on Neural Networks, Vol. 13 (3), pp.578–599, 2002.

[30] V. G. Kaburlasos, I. N. Athanasiadis, and P. A. Mitkas, “Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation”, International Journal of Approximate Reasoning, Vol. 45, pp. 152–188, 2007.

[31] E. Vrochidou, C. Lytridis, C. Bazinas, G. A. Papakostas, H. Wagatsuma, and V. G. Kaburlasos, “Brain Signals Classification Based on Fuzzy Lattice Reasoning”. Mathematics, Vol. 9, ID- 1063, 2021.

[32] V. G. Kaburlasos and V. Petridis, “Fuzzy Lattice Neuro computing (FLN) models”, Neural Networks, Vol. 13 (10), pp. 1145-1170, 2000.

[33] H.M. Liang and J.M. Mendel, “Interval type-2 fuzzy logic systems: theory and design”, Transactions on Fuzzy Systems, Vol. 8 (5), pp. 535–550, 2000.

[34] U. Urooj, B.A.S. Al-rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, “Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions”. Appl. Sci. Vol. 12, ID- 172, 2022.

[35] H. S. Talabani and H. M. T. Abdulhadi, “Bitcoin Ransomware Detection Employing Rule-Based Algorithms”, Science Journal of University of Zakho, Vol. 10, No. 1, pp. 5– 10, March-2022.

[36] Q.A. Al-Haija and A. A. Alsulami, “High Performance Classification Model to Identify Ransomware Payments for Heterogeneous Bitcoin Networks”. Electronics Vol. 10, ID-2113, 2021.

[37] A. Yazdinejad, H. HaddadPajouh, A. Dehghantanha, R. M. Parizi,. G. Srivastava, and M.-Y. Chen, “Cryptocurrency malware hunting: A deep Recurrent Neural Network approach”. Appl. Soft Comput. J. Vol. 96, ID-106630, 2020.

[38] O.M.K. Alhawi, J. Baldwin, and A. Dehghantanha, “Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection” In: Dehghantanha, A., Conti, M., Dargahi, T. (eds), Cyber Threat. Intell. Adv. Inf. Secur. Vol. 70, pp. 93–106, 2018.

[39] K. Kolesnikova, O. Mezentseva, and T. Mukatayev, “Analysis of Bitcoin Transactions to Detect Illegal Transactions Using Convolutional Neural Networks”. In Proceedings of the 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 28–30 April 2021; pp. 1–6

[40] C. Lee, S. Maharjan, K. Ko, J. Woo, and J.W.K. Hong, “Machine Learning Based Bitcoin Address Classification”. In Blockchain and Trustworthy Systems, BlockSys 2020. Communications in Computer and Information Science; Zheng, Z., Dai, H.N., Fu, X., Chen, B., Eds.; Springer: Singapore, Volume 1267, 2020.

[41] L. S. Burks, A. E. Cox, K. Lakkaraju, M. J. Boyd, and E. Chan, “Bitcoin Address Classification” (No. SAND2017-8407C); Sandia National Lab.(SNL-NM): Albuquerque, NM, USA, 2017.

[42] S. Homayoun, Ali Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, Kim-Kwang Raymond Choo, D. E. Newton, “DRTHIS: deep ransomware threat hunting and intelligence system at the fog layer”, Futur Gener Comput Syst Vol. 90, pp.94–104, 2019.

[43] J. Stiborek, T. Pevn ý, and M. Reh ́ak, “Multiple instance learning for malware classification”. Expert Syst Appl, Vol. 93, pp.346–57, 2018.

[44] S. K. Shaukat, and V. Ribeiro, “RansomWall : a layered defense system against cryptographic ransomware attacks using machine learning”, In: 10th International Conference on Communication Systems & Networks (COMSNETS); 2018. pp. 356–63.

[45] L. F. Maim ́o, A. H. Celdr ́an, A. L. P. G ́omez, Garc ía FJ Clemente, J. Weimer, and I. Lee, “Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments”, Sensors, Vol. 19(5), pp.1–31, 2019.

[46] S. D. Bolboacă, “Medical diagnostic tests: a review of test anatomy, phases, and statistical treatment of data”. Comput Math Methods Med Vol. 2019, ID-22, 2019, Hidawi.

[47] A. J. Vickers, B. Van Calster, and E. W. Steyerberg, “Net benefit approaches to the evaluation of prediction models, molecular markers, and diagnostic tests”, BMJ 2016; Vol. 352, pp.3–7, 2016.