

Ransomware Detection Using Stacked Autoencoder for Feature Selection

Mike Nkongolo Wa Nkongolo¹, Mahmut Tokmak²

¹Department of Informatics, University of Pretoria, South Africa

²Department of Management Information Systems, Bucak Zeliha Tolunay School of Applied Technology and Management, Mehmet Akif Ersoy University, Turkey

Article Info

Article history:

Received Sep 19, 2023

Revised Feb 14, 2024

Accepted Mar 10, 2024

Keyword:

Ransomware classification

Ransomware profiling

UGRansome dataset

Cryptology

Stacked autoencoder

Long Short-Term Memory

XGBoost

Intrusion detection

Signature attacks

Malware threats

Feature selection

Supervised learning

Deep learning

Ensemble learning

Machine learning

Autoencoder weights

Cyberintelligence

ABSTRACT

In response to the escalating malware threats, we propose an advanced ransomware detection and classification method. Our approach combines a stacked autoencoder for precise feature selection with a Long Short-Term Memory classifier which significantly enhances ransomware stratification accuracy. The process involves thorough preprocessing of the UGRansome dataset, training an unsupervised stacked autoencoder for optimal feature selection, and fine-tuning via supervised learning to elevate the Long Short-Term Memory model's classification capabilities. We meticulously analysed the autoencoder's learned weights and activations to pinpoint essential features for distinguishing 17 ransomware families from other malware and created a streamlined feature set for precise classification. Our results demonstrate the exceptional performance of the stacked autoencoder-based Long Short-Term Memory model across the 17 ransomware families. This model exhibits high precision, recall, and F1 score values. Furthermore, balanced average scores affirm its ability to generalize effectively across various malware types. To optimise the proposed model, we conducted extensive experiments, including up to 400 epochs, and varying learning rates and achieved an exceptional 98.5% accuracy in ransomware classification. These results surpass traditional machine learning classifiers. Moreover, the proposed model surpasses the Extreme Gradient Boosting (XGBoost) algorithm, primarily due to its effective stacked autoencoder feature selection mechanism and demonstrates outstanding performance in identifying signature attacks with a 98.5% accuracy rate. This result outperforms the XGBoost model, which achieved a 95.5% accuracy rate in the same task. In addition, a prediction of the ransomware financial impact using the proposed model reveals that while *Locky*, *SamSam*, and *WannaCry* still incur substantial cumulative costs, their attacks may not be as financially damaging as those of *NoobCrypt*, *DMALocker*, and *EDA2*.

Copyright © 2024 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Mike Nkongolo Wa Nkongolo

Department of Informatics

University of Pretoria

Lynnwood Road and Roper Street, Hatfield, Pretoria, South Africa

Email: mike.wankongolo@up.ac.za

1. INTRODUCTION

In today's digital age, ransomware has emerged as a significant threat to individuals and businesses alike [1]. Defined as a type of malicious software that encrypts valuable data and demands a ransom in exchange for its release, ransomware attacks have become increasingly prevalent and financially damaging [1, 2]. Recent incidents have resulted in staggering losses, reaching tens of millions of dollars for organisations [3]. In June 2022, the Serbian Republic Geodetic Authority, responsible for registering property rights,

experienced a ransomware attack. This attack disrupted regular services, making it difficult for citizens to make changes to real estate ownership in the registry [3]. Similar attacks have also been reported in neighboring countries. These include the Ministry of Agriculture of the Republic of North Macedonia, the Council of Ministers of Bosnia and Herzegovina, various public institutions in Albania, and the majority of critical governmental infrastructure in Montenegro [3]. South Africa, on the African continent, stands out as the country most impacted by ransomware and phishing emails [4]. The cybersecurity landscape in South Africa has exposed vulnerabilities in multiple sectors, resulting in a significant number of cyberattacks. Pieterse [5] highlights that public and private enterprises, as well as municipalities, are commonly targeted by ransomware attacks in South Africa. An example of this is the Department of Justice, which experienced its third ransomware attack in 2023, following a previous incident in 2020 [6]. These attacks have resulted in significant financial losses for various South African companies. Therefore, the urgency of tackling the global problem of classifying and detecting ransomware is evident, especially when considering the security of critical infrastructure [7]. There are several different types and variants of ransomware, each with its characteristics and behaviors (Table 1).

Table 1. Type of ransomware

| Type | Description |
|-----------|--|
| Crypto | Encrypts the victim's files, making them inaccessible. Victims are then presented with a ransom demand to obtain the decryption key. |
| Locker | Locker does not encrypt files but locks users out of their system or device. Victims are presented with a full-screen message demanding a ransom to regain access. |
| ScareWare | ScareWare displays fake security alerts or warnings, often claiming that the victim's computer is infected with malware or illegal content. Users are tricked into paying a fee for bogus security software or services. |
| Mobile | Mobile ransomware can lock a smartphone or tablet, encrypt files, or display threatening messages demanding payment. |
| MBR | Master Boot Record (MBR) infects a computer's MBR and prevents it from booting. Victims are then presented with a ransom message to unlock their system. |
| RaaS | Ransomware-as-a-Service (RaaS) allows cybercriminals to rent or purchase ransomware tools and services on the dark web. This enables even individuals with limited technical skills to launch ransomware attacks. |
| LeakWare | In addition to encryption, LeakWare also threatens to leak sensitive or confidential data unless the ransom is paid. This adds an extra layer of pressure on victims to comply. |
| WannaCry | WannaCry gained worldwide attention in 2017 when it infected hundreds of thousands of computers [16]. It exploited a Windows vulnerability to spread rapidly. |
| Ryuk | Ryuk is a targeted ransomware strain that primarily targets businesses and organisations. It often demands large ransoms. |
| NotPetya | This ransomware variant, which emerged in 2017 [17], was initially disguised as a ransomware attack but was later revealed to be a destructive wiper malware. |

In addition, the absence of readily accessible ransomware datasets within the current realm of intrusion detection poses a significant challenge to their accurate categorisation and detection [8, 9]. To address this limitation, our study uses the *UGRansome* dataset, a publicly accessible dataset created in [9]. This dataset was specifically designed to classify and understand ransomware [10–13]. In the age of big data, one crucial aspect of modern data analysis and machine learning implementation is the extraction of meaningful and representative features from complex or high-dimensional datasets [9, 14]. Among the various techniques available, stacked autoencoders (SAEs) have emerged as a potent tool for automating feature discovery [14, 15]. They enable the uncovering of intricate data structures and patterns. Grounded in the field of deep learning, SAEs provide an effective solution for addressing the challenge of representing high-dimensional data. They pave the way for improved predictive modeling, efficient dimensionality reduction, and insightful data interpretation [15].

1.1. Research Questions and Hypothesis

This section aims to outline the main research question, sub-research questions, and research hypothesis that will guide the research into harnessing SAEs and Long Short-Term Memory (*LSTM*) models for enhancing ransomware detection and classification using the *UGRansome* dataset.

1.1.1. Main research question

The main research question of this study can be stated as follows:

How can the integration of SAEs and LSTM models improve the classification and detection of ransomware using the UGRansome dataset?

1.1.2. Sub-research questions

The sub-research questions derived from the main research question are as follows:

What feature selection techniques can be incorporated within the SAE architecture to extract the most relevant and discriminative features from the UGRansome dataset?

How does the integration of feature selection techniques within the SAE architecture impact the effectiveness of ransomware detection and classification?

What temporal relationships within the feature space can be efficiently captured by the LSTM network to enhance ransomware detection?

What are the key advantages of the proposed approach for enhancing cybersecurity, particularly in the realm of ransomware recognition?

1.1.3. Research hypothesis

This section focuses on delineating the hypotheses derived from the main research question and sub-research questions to investigate the effectiveness of integrating SAEs and LSTM models for enhancing ransomware detection and classification using the UGRansome dataset:

Null hypothesis (H0): There is no significant difference in the effectiveness of feature selection techniques incorporated within the SAE architecture for extracting relevant and discriminative features from the UGRansome dataset. The integration of feature selection techniques within the SAE architecture does not significantly impact the effectiveness of ransomware detection and classification. There is no significant improvement in ransomware detection through efficient capturing of temporal relationships within the feature space by the LSTM network. There are no distinct advantages of the proposed approach for enhancing cybersecurity, particularly in the realm of ransomware recognition.

Alternative hypothesis (H1): Incorporating specific feature selection techniques within the SAE architecture will significantly improve the extraction of relevant and discriminative features from the UGRansome dataset. The integration of feature selection techniques within the SAE architecture will significantly enhance the effectiveness of ransomware detection and classification. Efficient capturing of temporal relationships within the feature space by the LSTM network will significantly enhance ransomware detection. The proposed approach will offer significant advantages for enhancing cybersecurity, particularly in the realm of ransomware recognition, compared to existing methods.

1.2. Research Limitation

One limitation of this study is the reliance on the UGRansome dataset, which, while comprehensive, may not fully capture the diversity and complexity of real-world ransomware attacks. This limitation could potentially affect the generalisability of the findings to other ransomware datasets and scenarios. Additionally, the study focuses on a specific set of feature selection techniques within the SAE architecture and overlooks alternative methods that could yield superior results. Furthermore, while the integration of SAE and LSTM models shows promise for ransomware detection and classification, the effectiveness of the proposed approach may be influenced by factors such as the quality and quantity of labeled data available for training the models.

1.3. Research Contribution

Our research endeavors to harness the combined power of SAEs and LSTM models to enhance the classification and detection of ransomware using the UGRansome dataset. The focus is on incorporating feature selection techniques within the SAE architecture to facilitate the extraction of the most relevant and discriminative features from the ransomware dataset. This approach selects ransomware input data while the subsequent LSTM network efficiently captures the temporal relationships within the feature space. The goal of this study is to contribute to the advancement of proactive and robust ransomware detection and classification strategies. The approach employed in this research holds several key advantages for enhancing cybersecurity, particularly in the realm of ransomware recognition. This research contributes significant insights into ransomware detection methodologies showcased by the SAE-LSTM model's superior performance over traditional machine learning algorithms such as XGBoost, and decision trees in terms of precision, recall, and F1 score. This performance superiority underscores the model's potential for accurately identifying ransomware instances and offers a promising avenue for zero-day exploit detection. Moreover, the study highlights the impactful role of feature selection using SAE, which notably enhances the model's precision and recall rates, emphasizing the criticality of feature selection in refining ransomware detection

algorithms. Additionally, the SAE-LSTM model demonstrates reduced misclassification rates, particularly in false positives and false negatives, compared to alternative algorithms like XGBoost and decision trees, thereby affirming its accuracy in correctly classifying ransomware instances. Furthermore, the model exhibits consistent effectiveness across 17 ransomware families, showcasing high precision, recall, and F1 score rates for different attack types, indicative of its versatility and robustness in detecting diverse ransomware variants. Lastly, the comprehensive prediction capabilities of the SAE-LSTM model are highlighted, with promising results in predicting signature, anomaly, and synthetic signature attacks with high accuracy, affirming its capacity to effectively identify a wide array of ransomware patterns, thus augmenting overall detection capabilities. This research paper is structured as follows: Section 2 offers a brief overview of related works, Section 3 outlines the research methodology, Section 4 presents the results and discussion, and Section 5 concludes the study.

2. RELATED WORKS

This related work section will provide context for the proposed research methodology. Feature selection using SAEs has been extensively studied in various domains. Wang et al. [18] proposed the use of Broad Autoencoder Features (BAF), which involves four inter-connected SAEs with different activation functions. The study proposes the BAF with four parallel connected SAEs using different activation functions and evaluates the performance of the BAF in terms of learned features using the Deep Neural Network (DNN). Another study by Kong et al. [19] explored the topic of feature extraction of load curves using an autoencoder network. Wang et al. [20] used a Stacked Supervised Auto-Encoder (SSAE) to train the deep network to obtain fault-relevant features. By stacking multiple supervised auto-encoders, high-level fault-relevant features are learned to improve the classification accuracy. In [21] the integration of SAE characteristics with wavelet-based and morphological fractal texture attributes was proposed for the classification of skin disorders. This approach achieved high accuracy in the classification task. Kim et al. [22] focused on proposing an SAE-based Convolutional Neural Network (CNN) model using discrete wavelet transform for feature extraction. The model aims to improve the accuracy of diagnosis by incorporating features from cutting force data, current signal, and coefficients of the discrete wavelet transform. In a paper by [23] a deep learning architecture with SAEs for intelligent malware detection based on Windows Application Programming Interface (API) was proposed. Similarly, [24] analysed the effectiveness of various deep learning and machine learning classifiers in detecting Android malware applications. The study uses different datasets and explores the use of Gabor filters and autoencoders to enhance classifier performance. In [25] a novel ensemble model, called Stacked Ensemble—Autoencoder (SEAE) for malware detection on the Internet of Things (IoT) domain was developed. The proposed model utilises three lightweight neural network models trained on essential features extracted from the *Mallmg* dataset. The model demonstrates high accuracy (99.43%) in classifying malware images and outperforms existing approaches. In summary, the studies discussed in this section emphasise the benefits of using SAEs for feature selection across different domains and tasks. In Section 4, we conduct a comparative analysis of these studies with the proposed SAE-LSTM methodology.

2.1. Long Short-Term Memory

LSTM networks have emerged as powerful tools in cybersecurity, particularly in ransomware detection. LSTM networks, known for their ability to capture long-term dependencies in sequential data, offer a promising approach for analysing the complex and dynamic nature of ransomware behaviors. LSTM's capabilities can be used in modeling sequential data. In this context, they have become indispensable tools for cybersecurity practitioners seeking to classify network threats. In comparison to the papers discussed in Section 2, our research introduces a unique approach that combines feature selection using SAE and classification with an LSTM model which resulted in improved ransomware classification accuracy. The process includes preprocessing the UGRansome dataset, training an unsupervised SAE model for feature extraction, and then fine-tuning the LSTM algorithm with supervised learning to enhance its classification capabilities. LSTM is a type of Recurrent Neural Network (RNN) architecture that is used for processing sequential data [26]. Unlike traditional RNNs, LSTM is designed to capture long-term dependencies effectively. It achieves this by using a memory cell, which has three components: an *input gate*, a *forget gate*, and an *output gate*. The input gate decides how much new information should be stored in the memory cell, while the forget gate determines what information should be forgotten. The output gate controls the amount of information that is output from the memory cell to the next step. By using these gates, LSTM can process sequential data more accurately to capture long-term dependencies [26]. LSTM networks have shown considerable promise in detecting malware. Researchers have invested substantial effort into optimising LSTM hyperparameters specifically for the design of Intrusion Detection Systems (IDS) [26, 27]. These endeavors have led to the exploration of various LSTM configurations, revealing that the importance of hyperparameters for LSTM in IDS differs significantly from

their roles in language models. The intricate interplay between these hyperparameters has a pronounced impact on their relative significance. Taking this interplay into account, the hierarchy of importance for LSTMs in IDS becomes clear, with batch size emerging as the most critical factor, followed by dropout ratio and padding [26]. Additionally, innovative LSTM models have been proposed for the creation of systems focusing on behavioral language for malware detection [27]. These models have demonstrated impressive performance metrics, including high accuracy values and specificity when tested on unfamiliar attack datasets. Another approach involves leveraging LSTM in conjunction with word embedding and attention mechanisms to effectively represent and classify malware files [26]. This strategy has yielded remarkable results, achieving high accuracy and F1 scores [27]. Fang et al. [28] conducted a study where they introduced a novel method for zero-day exploit recognition using LSTM. Their model is designed specifically for identifying malicious JavaScript code injected into web pages [29] by extracting features from the semantic level of bytecode and optimising word vectorization techniques. The findings of their research reveal that the LSTM-based detection model outperforms existing models that rely on tree-based algorithms. In addition, Roberts and Nair [30] propose a neural architecture that addresses the problem of anomaly detection in discrete sequence datasets. Their approach involves modifying the LSTM autoencoder and incorporating an array of one-class support vector machines (SVM) to detect anomalies within sequences. This method demonstrates improved stability and performs better compared to traditional LSTM-based anomaly detection systems. One limitation of this approach is that it requires a labeled dataset for training the one-class SVM, which can be challenging to obtain in certain domains.

2.2. Ransomware Detection and Classification using Machine Learning

In recent years, the field of ransomware detection and classification has seen significant progress driven by advancements in machine learning and deep learning methodologies. This section delves into various approaches and techniques employed to combat ransomware threats. One notable aspect is the utilisation of behavioral analysis, where ransomware families are scrutinised for common traits like payload persistence and obfuscation techniques [47]. Additionally, machine learning algorithms such as random forest, decision tree, logistic regression, naïve Bayes, and neural networks have been harnessed to classify ransomware based on selected features [47]. Frameworks like Biflow and Droid-NNet have also emerged, offering novel methodologies for detecting and classifying ransomware [47]. Furthermore, techniques such as signature parsing, n-gram analysis, and LSTM networks have been leveraged to classify ransomware based on behavioral patterns, API calls, and network traffic. These advancements underscore the importance of integrating advanced machine learning approaches to enhance ransomware detection and prevention strategies. Continued innovation in this field is crucial for staying ahead of evolving ransomware threats and ensuring robust cybersecurity measures.

3. RESEARCH METHOD

This section addresses the limitations of using legacy datasets for ransomware detection as discussed in Section 1 and Section 2. These datasets often contain outdated or insufficiently diverse attack data. In response to this limitation, our study leverages a recently designed ransomware dataset tailored specifically for ransomware detection. This dataset serves as the foundation for evaluating the efficacy of the proposed SAE-LSTM model in detecting ransomware attacks. In 2021, Nkongolo et al. [9] introduced the UGRansome dataset as a valuable resource for detecting ransomware attacks and zero-day exploits [11, 31]. This dataset represents a significant contribution to the field of cybersecurity research. The dataset was created by combining features of the UGR'16 and ransomware datasets [9]. A fuzzy merging algorithm was employed to amalgamate the most similar features from these datasets. This resulted in the formation of the final UGRansome dataset [9]. Before the fuzzy merging process, Principal Component Analysis (PCA) was utilised to retrieve features. A script was executed for the *fuzzy merging* process, implemented on the Linux platform. To safeguard privacy, sensitive features such as IP addresses and ransomware transaction links were encoded using cryptographic transformation [9]. Despite its strengths, it is important to acknowledge that the UGRansome dataset may exhibit redundancy and require thorough data cleaning and processing. This dataset has been utilised in previous studies which underscore its value in advancing cybersecurity research [11, 32, 8, 31]. The UGRansome dataset stands out for its inclusion of zero-day exploits including EDA2, SamSam, JigSaw, NerisBonet, advanced persistent threats (APT), and TowerWeb, which have not been explored previously. Each attack in the dataset is labeled into predictive/target variables such as Anomaly (A), Signature (S), and Synthetic Signature (SS) [11, 32]. In the context of ransomware classification, the target variable typically refers to the label assigned to each instance of data, indicating whether it represents:

Anomaly (A): Instances that are identified as anomalous or suspicious behavior, indicating potential ransomware activity.

Signature (S): Instances that match known signatures or patterns of ransomware, suggesting a confirmed ransomware attack.

Synthetic Signature (SS): Instances that are artificially generated to simulate ransomware behavior, often used for training and testing machine learning models. In addition, the UGRansome dataset covers a range of malware such as Blacklist, Scan, and Spam [8, 31]. When presenting the size of the UGRansome, it is important to provide clear and concise information that provides a comprehensive understanding of its scale. We can effectively present the size of this dataset as follows:

Number of records: The experimental dataset contains a total of 149,043 instances. Within this dataset, there are over 60,000 instances labeled as Signature (S), around 40,000 instances labeled as Anomaly (A), and Synthetic Signature (SS) records (<https://www.kaggle.com/dsv/7172543>).

Number of attributes: The dataset contains 14 attributes or columns arranged as 'Time', 'Protocol', 'Flag', 'Family', 'Clusters', 'SeedAddress', 'ExpAddress', 'BTC', 'USD', 'Netflow_Bytes', 'IPaddress', 'Threats', 'Port', and 'Prediction' (Table 2) (<https://www.kaggle.com/dsv/7172543>). In addition, there are 17 ransomware types detected in this dataset (Figure 10). The data characteristics are as follows:

Time span: Data was collected over a period of 9 months [9].

Data format: The dataset is structured as a CSV file (<https://www.kaggle.com/dsv/7172543>).

Storage size: The total file size of the dataset is 10.0 MB (<https://www.kaggle.com/dsv/7172543>).

To understand the dataset in more detail, we refer to Table 2, which highlights its attributes and characteristics.

Table 2. The UGRansome dataset

| Column | Attribute | Meaning | Type | Example |
|--------|---------------|---|-------------|---------------------|
| 1 | Time | Timestamp of network attacks | Numeric | 45s |
| 2 | Protocol | Network protocol | Categorical | ICMP, UDP, TCP |
| 3 | Flag | Network connection status | Categorical | SYN, ACK |
| 4 | Family | Ransomware type | Categorical | WannaCry, Crypto |
| 5 | Cluster | Malware clusters or groups | Numeric | 1-12 |
| 6 | Seed address | Formatted ransomware links | Categorical | 18Syst8y345 |
| 7 | Exp address | Original ransomware links | Categorical | Syst345y18 |
| 8 | BTC | Financial damages in Bitcoin caused by ransomware/malware | Numeric | 80.90 BTC |
| 9 | USD | Financial damages in USD caused by ransomware/malware | Numeric | 814,678\$ |
| 10 | Netflow bytes | Bytes transferred in network flow | Numeric | 987,987 |
| 11 | IP address | Connection identification | Categorical | A, B, and C |
| 12 | Threats | Malware | Categorical | Spam, and Blacklist |
| 13 | Port | Network port number | Numeric | 5062 |
| 14 | Prediction | Target variable | Categorical | Anomaly (A) |

3.1. Stacked Autoencoder for Feature Selection

Various deep learning and machine learning methods such as SAE and recursive feature elimination (RFE) can be employed with the UGRansome dataset. SAEs are a versatile type of neural network architecture utilised for feature extraction and dimensionality reduction in various domains [33]. A SAE is composed of multiple layers, with each layer learning to reconstruct the input data (Figure 1). This algorithm stacks these layers and allows the network to learn complex features from the data [33]. The key idea is to encode the input data into a lower-dimensional representation, and then decode it back into its original form (Figure 1) [33]. This process helps extract meaningful features and patterns from the data, making it useful for tasks like dimensionality reduction and feature learning. SAEs have found applications in biometrics recognition, image recognition, natural language processing, and automatic speech recognition [33]. The stacked nature of autoencoders arises from their composition, which includes multiple layers of autoencoders (Figure 1). Each layer is tasked with reconstructing the output of the preceding layer. Training SAEs involves two critical steps: *unsupervised pre-training* and *supervised fine-tuning* [33]. In the unsupervised pre-training phase, individual layers within the network are trained using autoencoders, which specialise in learning internal data representations. These representations serve to initialise the network weights and enhance its generalisation capabilities. Subsequently, in the supervised fine-tuning stage, the pre-trained layers are assembled and jointly trained using labeled data. This approach has been reported to achieve exceptional accuracy rates [33].

3.2. SAE Architecture

Figure 1 illustrates the proposed SAE architecture. This architecture starts by taking the UGRansome dataset as input features. It then encodes and decodes UGRansome using an SAE scheme which reconstructs important features and passes them to the LSTM algorithm for classification purposes. This classification

assigns the Anomaly (A), Synthetic Signature (SS), or Signature (S) categories to these features. The A and SS categories represent zero-day exploits, while the S category represents well-known attacks [9].

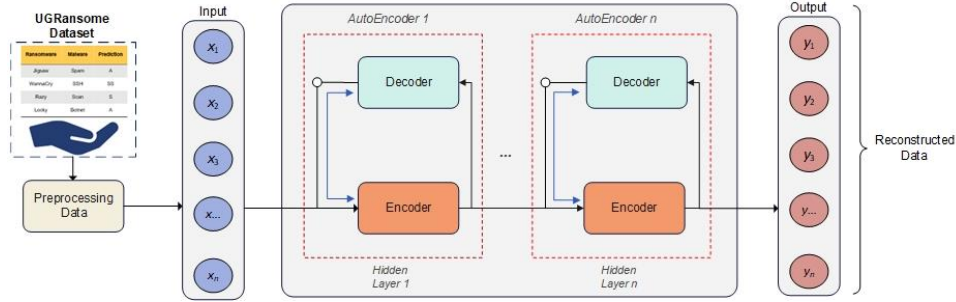


Figure 1. SAE architecture

3.3. Unsupervised pre-Training Objective Function for Layer l

The unsupervised pre-training objective function for layer l aims to optimise the encoding and decoding process of the SAE [33]. It involves reconstructing the input data by optimising/minimising the reconstruction error. The objective function for unsupervised pre-training of layer l is given in (1).

Where:

- W represents the weights of layer l
- b denotes biases of layer l
- x illustrates input data for the i -th training example in layer l
- m represents the number of training examples
- \hat{x} denotes the output data specific to each training example

$$\min_{W^{(l)}, b^{(l)}} \frac{1}{m} \sum_{i=1}^m \|x^{(l)}(i) - \hat{x}^{(l)}(i)\|^2 \quad (1)$$

3.4. Supervised fine-Tuning

Supervised fine-tuning involves adjusting the parameters of the model based on labeled data [35]. It aims to improve the model's performance on classification tasks by iteratively updating its weights and biases to minimise a predefined loss function [33, 35]. After unsupervised pre-training, the layers are stacked together to form the full SAE. The network is then trained using a supervised *loss function*, typically a classification loss, with labeled data [35]. The mathematical formulation of the supervised cross-entropy loss function is given in (2).

$$\mathcal{L}_{\text{supervised}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(p_{ij}) \quad (2)$$

Where:

- N represents the number of training examples
- C denotes the number of classes
- y is an indicator for correct classification
- p represents predicted probability

3.5. Recurrent Neural Network

A recurrent neural network (RNN) is a type of neural network designed to work with sequential data. It processes input sequences one element at a time by maintaining an internal state to capture information about previous elements (Figure 2). This allows RNNs to model temporal dependencies in data, making them well-suited for tasks such as time series prediction, natural language processing, and speech recognition [34]. As such, an RNN can be thought of as a variation of the feedforward Neural Network (FFNN) that introduces a recurrent structure within its network [34, 36] (Figure 2). While the FFNN comprises multiple layers with unidirectional connections, RNN establishes connections from each neuron to itself. This self-connection mechanism allows RNN to retain previous inputs which could potentially influence the network's output [36]. In RNN, the inference process is similar to that of the FFNN, completed through forward propagation. Training in RNN is accomplished using *backpropagation* through time, where the weights are updated based on the gradient [35].

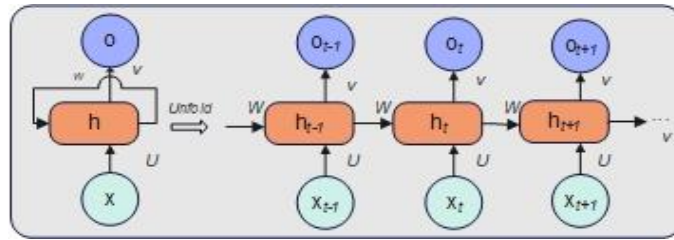


Figure 2. RNN architecture

However, RNNs face limitations in handling long-term dependencies due to the *vanishing gradient* problem [36], where gradients diminish exponentially over time during backpropagation, leading to ineffective learning of long-range dependencies. Unlike traditional RNNs, LSTM networks overcome this limitation by incorporating *gated units* which regulate information flow and enable the network to retain and selectively update information over multiple time steps. This architecture allows LSTMs to effectively capture and learn long-term dependencies in sequential data, making them more suitable for tasks requiring memory of past information over extended periods [36].

3.6. RNN and LSTM

To address RNN issues, the LSTM deep learning algorithm was developed by Hochreiter and Schmidhuber in 1997 as a variant of the RNN model [35, 36]. LSTM introduces the concept of *memory cells* for its nodes, enabling the linkage of prior data information to the present nodes. Each LSTM node incorporates three gating mechanisms: an *input gate*, a *forget gate*, and an *output gate* (Figure 3).

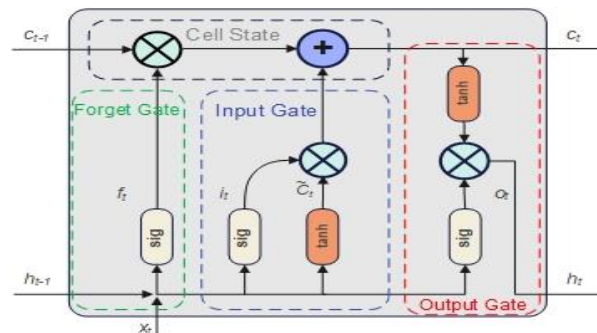


Figure 3. LSTM architecture

The key components of the LSTM gating mechanisms can be defined as follows:

- i (input gate) controls the flow of new information into the memory cell (3).
- f (forget gate) controls the flow of information to forget from the previous memory cell state (4).
- o (output gate) controls the output from the memory cell (5).
- C represents the cell state (6).
- H illustrates the hidden state (7).

The LSTM equations for these gating mechanisms are as follows:

$$\text{Input Gate: } i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\text{Forget Gate: } f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \quad (4)$$

$$\text{Output Gate: } o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \quad (5)$$

$$\text{Cell State Update: } c_t = f_t \times c_{t-1} + i_t \times \tanh(W_c \times [h_{t-1}, x_t] + b_c) \quad (6)$$

$$\text{Hidden State Update: } h_t = o_t \times \tanh(c_t) \quad (7)$$

3.7. Extreme Gradient Boosting

Gradient boosting is a machine learning technique used for regression and classification tasks [37]. It works by combining multiple weak learners, typically decision trees, into a single strong learner. The algorithm iteratively builds a series of trees, each one focusing on the mistakes made by the previous trees [37]. The predictions of these trees are used by gradient boosting to produce a highly accurate model. This model is known for its robustness and ability to handle complex datasets with high dimensionality [37]. In this research, we compare the performance of the proposed *SAE-LSTM* model with that of the Extreme Gradient Boosting (*XGBoost*) using the *UGRansome* dataset. *XGBoost* is also a powerful and efficient machine learning algorithm used for both regression and classification tasks [37]. It belongs to the ensemble learning category. *XGBoost* is known for its high predictive accuracy and is widely used in various data science and machine learning competitions [37]. The algorithm aims to find an optimal model by minimising a loss function that measures the difference between predicted values and actual target values [37] and builds a strong predictive model by iteratively combining multiple weak decision trees. This algorithm uses the following concepts:

$$\text{Objective function: } \text{obj}(\theta) = L(\theta) + \Omega(\theta) \quad (8)$$

This is the overall function that *XGBoost* aims to optimise during training in (8) [37]. It is a combination of two main parts: the *loss function* and the *regularization* term. The goal of *XGBoost* is to find the best values of model parameters that minimise the objective function [37].

$$\text{Loss function: } L(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) \quad (9)$$

The loss function shown in (9) measures the discrepancy between the actual target values y and the predicted values $\hat{y}_i^{(t)}$ generated by the current iteration of the model [37]. The loss function quantifies how well the model is performing on the training data. Its objective is to minimise the loss by adjusting the model's parameters [37].

$$\text{Regularization term: } \Omega(\theta) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (10)$$

The regularization function illustrated in (10) is a technique used to prevent overfitting, which occurs when a model fits the training data too closely and does not generalise well to new data [37]. In *XGBoost*, there are two components to the regularization term:

γT : This term discourages the model from creating complex rules.

$\frac{1}{2} \lambda \sum_{j=1}^T w_j^2$: This term discourages the model from assigning excessively large weights. The function in (11) computes the predicted value for a specific data point (x) at a given iteration (t) of the boosting process [37].

$$\hat{y}_i^{(t)} = \phi(x_i) = \sum_{k=1}^K f_k(x_i) \quad (11)$$

It represents the sum of predictions from individual trees $f(x)$ in the model. As boosting iterations progress, more trees are added, and the prediction is updated. In summary, *XGBoost* seeks to find the best model parameters by minimising a combination of two factors: how well the model fits the training data (*loss function*) and how complex the model is (*regularization term*). The prediction function $\hat{y}_i^{(t)}$ represents the model's output for a specific data point at a given iteration. The goal is to iteratively improve the model by adjusting its parameters and thereby reducing the overall objective function.

3.8. Performance Evaluation

Performance evaluation is the process of assessing a machine learning model's effectiveness and efficiency [9]. It involves measuring the model's ability to make accurate predictions on unseen data and comparing its performance against predefined metrics [12]. These metrics may include accuracy, precision, recall, F1 score, area under the ROC curve (AUC-ROC), and mean squared error (MSE), among others. Performance evaluation helps researchers and practitioners determine the strengths and weaknesses of a model and identify areas for improvement. Hence, the training and testing evaluation performance of the proposed *SAE-LSTM* for ransomware classification is crucial. Several metrics were used to assess the effectiveness of

the proposed model, including accuracy, precision, recall (sensitivity), and the F1 score [12, 38, 39]. Accuracy is a fundamental metric that quantifies the proportion of correctly classified instances out of the total instances [9]. Mathematically, accuracy is expressed as in (12):

$$Accuracy = \frac{\text{Number of Correctly Classified Instances}}{\text{Total Number of Instances}} \times 100\% \quad (12)$$

Precision assesses the accuracy of positive predictions among the instances that are predicted as positive [12]. It is defined as in (13):

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (13)$$

Recall, also known as sensitivity or true positive rate, measures the proportion of true positive values correctly identified by the model [9]. It is formally defined as in (14):

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (14)$$

The F1 score is a composite metric that combines precision and recall [7, 40]. It provides a balance between these two metrics and is mathematically defined as in (15):

$$F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)$$

In addition to these metrics, a confusion matrix was used to provide a more detailed evaluation of the proposed model performance [12]. The confusion matrix summarises the classification results into four categories:

- True Positives (TP). These are instances correctly classified as positive.
- True Negatives (TN). Represents instances correctly classified as negative.
- False Positives (FP). Denotes instances incorrectly classified as positive.
- False Negatives (FN). Illustrates instances incorrectly classified as negative.

The confusion matrix (Figure 4) allows for a deeper understanding of the SAE-LSTM model's performance, especially in scenarios where class imbalances exist.

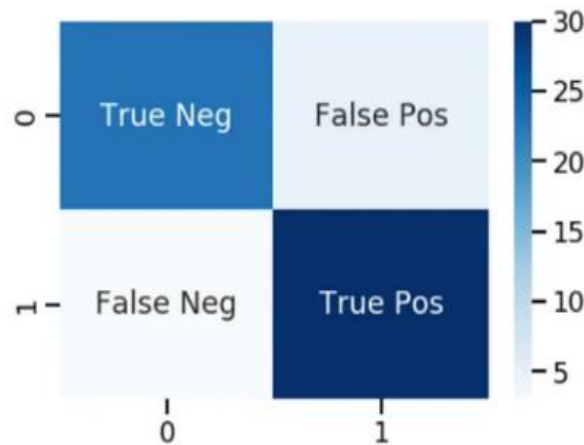


Figure 4. Confusion matrix

The methodological approach employed in this research is visually depicted in Figure 5 and Algorithm 1. This methodology commences with the UGRansome dataset which undergoes preprocessing and normalisation using Python data encoding techniques like label encoder and standard scaler (Figure 5). The pre-processed UGRansome data is then fed into the SAE for feature selection (Figure 5), which extracts the most significant features such as ransomware families. These features are split into an 80% training set and a 20% testing set using cross-validation (Figure 5). The testing set is subsequently utilised by the LSTM classifier

to predict the category of extracted ransomware, resulting in a tripartite prediction comprising S, SS, and A attacks (Figure 5). The methodology employed in this research is designed to effectively detect and classify ransomware attacks using a combination of data preprocessing, feature extraction, SAE, and LSTM techniques (Figure 5).

3.8.1. Data preprocessing

The methodology begins with the UGRansome dataset, a crucial resource for training and evaluating the ransomware detection model [9, 42]. To ensure the data is suitable for analysis, preprocessing steps are performed. This includes handling missing values, encoding categorical variables using techniques like label encoding, and scaling numerical features using methods such as standard scaling. The choice of *Python* for data encoding aligns with its widespread use in machine learning and data analysis tasks which provide a flexible and efficient environment for data manipulation.

3.8.2. Feature extraction with SAE

Once the data is pre-processed, it is passed through the SAE-based feature extraction component (Figure 5). SAEs are chosen for their ability to automatically learn and extract meaningful features from raw data. They can reconstruct input data through a series of encoding and decoding layers and capture hierarchical representations of the data. The rationale behind using SAEs lies in their effectiveness in capturing complex patterns and relationships within the UGRansome dataset, particularly in identifying key features indicative of ransomware families.

3.8.3. Data splitting and cross-validation

After feature extraction, the dataset is split into training and testing sets. The common practice of an 80% training and 20% testing split is employed. Additionally, cross-validation is utilised to further validate the model's performance. This technique helps mitigate overfitting by iteratively splitting the data into training and validation sets [9], providing a more robust estimate of the model's generalisation performance.

3.8.4. Classification with LSTM classifier

The pre-processed and split dataset is then used to train an LSTM classifier. LSTM networks are well-suited for sequential data like time series and text data, making them a suitable choice for detecting patterns in the sequential nature of ransomware attacks. The LSTM classifier predicts the category of extracted ransomware, with classifications falling into three categories: S (signature attacks), SS (synthetic signature attacks), and A (anomaly attacks) (Figure 5). In this approach, the experiments were executed using Python version 3.10.12. The training and testing phases of the proposed data encoding, normalisation, SAE, and LSTM model were carried out using the Google Colaboratory cloud system. This platform offers convenient access to a wide array of Python libraries and services at no cost.

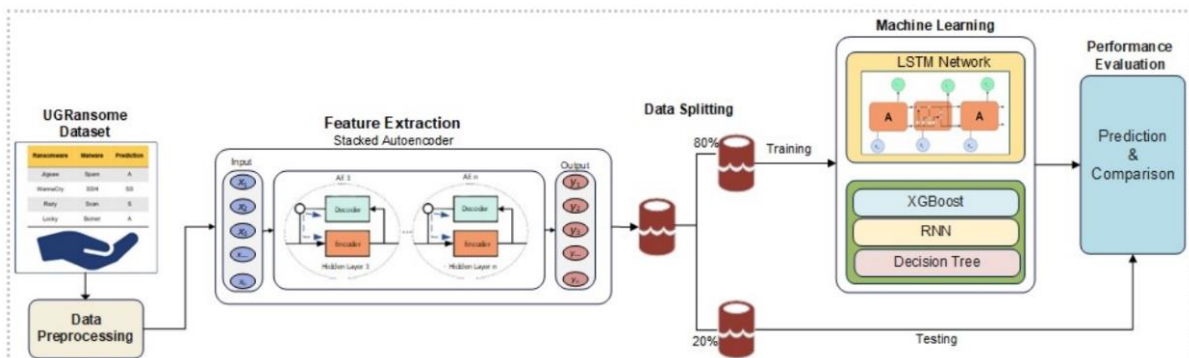


Figure 5. Proposed SAE-LSTM framework

To enhance algorithmic execution speed, *Nvidia CUDA* technology within the *Colab* environment was utilised. Various essential tasks, including file uploading, data pre-processing, and data frame setup were accomplished using *Python* libraries such as *numpy*, *pandas*, *statistics*, *sklearn*, *matplotlib.pyplot*, and *seaborn*. To implement the recommended SAE-LSTM architecture, the *Python TensorFlow Keras* library was employed.

Algorithm 1 SAE-LSTM Training

```

1:   Initialise the encoder and decoder neural network models
2:   Define a loss function  $L$  (e.g., MSE)
3:   Define an optimiser (e.g., Stochastic Gradient Descent)
4:   for each training epoch do
5:       for each training batch do
6:           Forward pass:
7:               Pass the input data  $x_t$  through the encoder to obtain encoded features
8:                $h_t = g(W_i \times [h_{t-1}, x_t] + b_i)$ 
9:               Pass the encoded features  $h_x$  through the decoder to obtain decoded features
10:               $\hat{x}_t = \sigma(W_o \times h_t + 0)$ 
11:              Compute the loss:  $L(x_t, \hat{x}_t)$ 
12:              Backward pass:
13:                  Calculate gradients using backpropagation
14:                  Update the weights and biases  $W_i, W_o, b_i, b_o$  using the optimiser
15:                  Compute the average loss for the epoch
16:                  if the average loss is below a predefined threshold or after a fixed number of
17:                  epochs then
18:                      break ▷ Training convergence criteria met
19:          Use the trained autoencoder for feature selection
20:          Extract the encoded features  $h_t$  from the encoder
21:          These encoded features can be used as selected features for LSTM classification

```

The specified SAE architecture comprised three encoders with 75, 50, and 13 layers, respectively, and three corresponding decoders with 50, 75, and 13 layers (Table 3).

Table 3. SAE layer and parameter

| Layer (type) | Output shape | Param # |
|-----------------------|--------------|---------|
| input_1 (Input Layer) | (None, 13) | 0 |
| dense | (None, 75) | 1,058 |
| dense_1 | (None, 50) | 3,800 |
| dense_2 | (None, 13) | 663 |
| dense_3 | (None, 50) | 700 |
| dense_4 | (None, 75) | 3,825 |
| desnse_5 | (None, 13) | 988 |
| Total params | | 11,026 |

The activation function was configured as *relu*, the optimizer as *Adam*, the loss as *MSE*, and the number of epochs as 50 (Table 3). The constructed LSTM network consisted of 3 layers, each containing 168 neurons (Table 4). The loss parameter was set to sparse categorical cross-entropy, the optimiser to *Adam*, and the number of epochs to 400.

Table 4. LSTM layer and parameter

| Layer (type) | Output shape | Param # |
|---------------------|--------------|---------|
| Lstm_3 (LSTM) | (None, 168) | 122,304 |
| dense_21 (Dense) | (None, 3) | 507 |
| Total params | | 122,811 |

4. RESULTS AND DISCUSSION

In this section, we present the results obtained using our proposed SAE-LSTM model. The results begin by providing a comprehensive discussion of various experimental facets. This includes (i) the data pre-

processing and encoding procedures, (ii) the results of feature extraction using SAE, (iii) the cross-validation process involving data splitting, (iv) the performance of the LSTM classifier, and (v) the predictive modeling of ransomware categorised into S, A, and SS (Figure 5).

4.1. Data Encoding and pre-Processing

Figure 6 provides an overview of the UGRansome statistics. The original UGRansome consisted of 207,533 features, with 58,491 redundant patterns that account for 28.2% of the dataset (Figure 6). The graph (Figure 6) illustrates the contrast in network flow between the duplicate and cleaned datasets. The duplicate data appears sparser, with a significantly lower density around the 500 NetFlow bytes mark. In contrast, the cleaned data exhibits a more positive skew. Within the scope of this study, the *sklearn* preprocessing library played a pivotal role in the conversion of categorical attributes into numeric representations across multiple columns within the UGRansome dataset (Figure 7).

| Dataset Statistics | | Dataset Statistics | |
|----------------------------|--|----------------------------|--|
| Number of Variables | 14 | Number of Variables | 14 |
| Number of Rows | 207533 | Number of Rows | 149042 |
| Missing Cells | 0 | Missing Cells | 0 |
| Missing Cells (%) | 0.0% | Missing Cells (%) | 0.0% |
| Duplicate Rows | 58491 | Duplicate Rows | 0 |
| Duplicate Rows (%) | 28.2% | Duplicate Rows (%) | 0.0% |
| Total Size in Memory | 106.9 MB | Total Size in Memory | 78.0 MB |
| Average Row Size in Memory | 540.2 B | Average Row Size in Memory | 548.5 B |
| Variable Types | Numerical: 4 Categorical: 9 GeoGraphy: 1 | Variable Types | Numerical: 4 Categorical: 9 GeoGraphy: 1 |

(a) Dataset with redundancy

(b) Dataset without redundancy

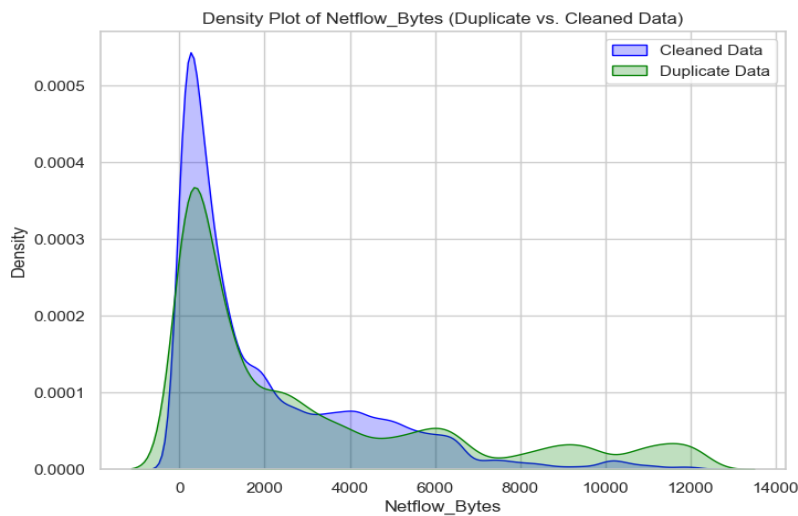


Figure 6. Dataset characteristics

The statistical analysis includes boxplots for each feature, where categorical values were converted to numerical values to enable their representation (Figure 7). Notable features with outliers include timestamps, clusters, BTC, USD, and Netflow_bytes. Among these, the BTC feature stands out with the highest number of outliers. The boxplots provide information on various statistical parameters for each feature, such as minimum and maximum values, skewness, first and third quartiles, and medians. To eliminate redundancy, the SAE ignored duplicate rows during the feature selection process (Figure 6). We then employed a methodology known as label encoding to transform UGRansome data. The primary objective underlying this encoding strategy was to render the dataset compatible with machine learning algorithms that mandate numeric inputs for their operation.

| | Time | Protocol | Flag | Ransomware | Clusters | SeedAddress | ExpAddress | BTC | USD | Netflow_Bytes | IPAddress | Malware | Port | Prediction |
|--------|------|----------|------|------------|----------|-------------|------------|------|------|---------------|-----------|---------|------|------------|
| 0 | 40 | TCP | A | WannaCry | 1 | 1DA11mPS | 1BonuSr7 | 1 | 504 | 8 | A | Bonet | 5061 | SS |
| 1 | 30 | TCP | A | WannaCry | 1 | 1DA11mPS | 1BonuSr7 | 1 | 508 | 7 | A | Bonet | 5061 | SS |
| 2 | 20 | TCP | A | WannaCry | 1 | 1DA11mPS | 1BonuSr7 | 1 | 512 | 15 | A | Bonet | 5061 | SS |
| 3 | 57 | TCP | A | WannaCry | 1 | 1DA11mPS | 1BonuSr7 | 1 | 516 | 9 | A | Bonet | 5061 | SS |
| 4 | 41 | TCP | A | WannaCry | 1 | 1DA11mPS | 1BonuSr7 | 1 | 520 | 17 | A | Bonet | 5061 | SS |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 149037 | 33 | UDP | AP | TowerWeb | 3 | 1AEoiHYZ | 1SYSTEMQ | 1010 | 1590 | 3340 | A | Scan | 5062 | A |
| 149038 | 33 | UDP | AP | TowerWeb | 3 | 1AEoiHYZ | 1SYSTEMQ | 1014 | 1596 | 3351 | A | Scan | 5062 | A |
| 149039 | 33 | UDP | AP | TowerWeb | 3 | 1AEoiHYZ | 1SYSTEMQ | 1018 | 1602 | 3362 | A | Scan | 5062 | A |
| 149040 | 33 | UDP | AP | TowerWeb | 3 | 1AEoiHYZ | 1SYSTEMQ | 1022 | 1608 | 3373 | A | Scan | 5062 | A |
| 149041 | 33 | UDP | AP | TowerWeb | 3 | 1AEoiHYZ | 1SYSTEMQ | 1026 | 1614 | 3384 | A | Scan | 5062 | A |

149042 rows x 14 columns

| | Time | Protocol | Flag | Ransomware | Clusters | SeedAddress | ExpAddress | BTC | USD | Netflow_Bytes | IPAddress | Malware | Port | Prediction |
|--------|------|----------|------|------------|----------|-------------|------------|------|------|---------------|-----------|---------|------|------------|
| 0 | 40 | 1 | 0 | 16 | 1 | 2 | 2 | 1 | 504 | 8 | 0 | 1 | 5061 | 2 |
| 1 | 30 | 1 | 0 | 16 | 1 | 2 | 2 | 1 | 508 | 7 | 0 | 1 | 5061 | 2 |
| 2 | 20 | 1 | 0 | 16 | 1 | 2 | 2 | 1 | 512 | 15 | 0 | 1 | 5061 | 2 |
| 3 | 57 | 1 | 0 | 16 | 1 | 2 | 2 | 1 | 516 | 9 | 0 | 1 | 5061 | 2 |
| 4 | 41 | 1 | 0 | 16 | 1 | 2 | 2 | 1 | 520 | 17 | 0 | 1 | 5061 | 2 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 149037 | 33 | 2 | 2 | 15 | 3 | 1 | 6 | 1010 | 1590 | 3340 | 0 | 6 | 5062 | 0 |
| 149038 | 33 | 2 | 2 | 15 | 3 | 1 | 6 | 1014 | 1596 | 3351 | 0 | 6 | 5062 | 0 |
| 149039 | 33 | 2 | 2 | 15 | 3 | 1 | 6 | 1018 | 1602 | 3362 | 0 | 6 | 5062 | 0 |
| 149040 | 33 | 2 | 2 | 15 | 3 | 1 | 6 | 1022 | 1608 | 3373 | 0 | 6 | 5062 | 0 |
| 149041 | 33 | 2 | 2 | 15 | 3 | 1 | 6 | 1026 | 1614 | 3384 | 0 | 6 | 5062 | 0 |

149042 rows x 14 columns

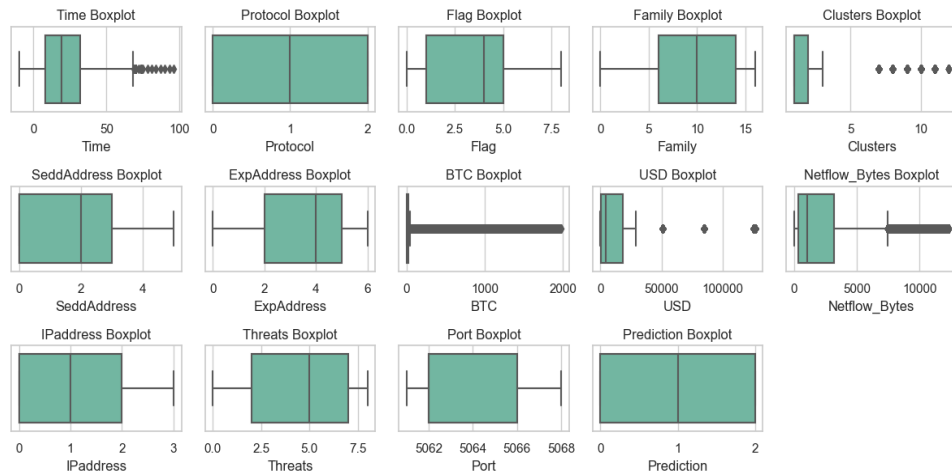


Figure 7. Comparison of characteristics in the pre-processed dataset

Using this process, categorical variables were effectively transformed into numerical equivalents, thereby rendering them computable to various modeling and analytical techniques. This enhanced dataset, consisting of numeric representations, becomes a valuable asset in the context of LSTM classification. The dataset initially contained zero and negative values in the timestamp feature, which represents network flow (Figure 8). Given that timestamps cannot logically be zero or negative in this context, they were removed to ensure data accuracy and quality. After this cleansing, the remaining timestamps had an average duration of approximately 21 seconds across the dataset, indicating prolonged dataflows, possibly associated with zero-day threats (Figure 8). The standard deviation highlights the variability in these timestamp values. Similarly, some columns show a smaller standard deviation value because they have smaller ranges (Figure 8). Other columns have a much higher standard deviation. While most of the values are fairly low, the outliers are larger numbers.

4.2. Feature Selection Results for Ransomware Classification

The initial phase of the analysis involved an examination of the distribution of ransomware instances selected by the SAE. It was observed that Locky, SamSam, and WannaCry exhibited the highest frequency of

occurrences, whereas EDA2 and DMALocker occupied a middle ground, with NoobCrypt registering a relatively lower count (Figure 9).

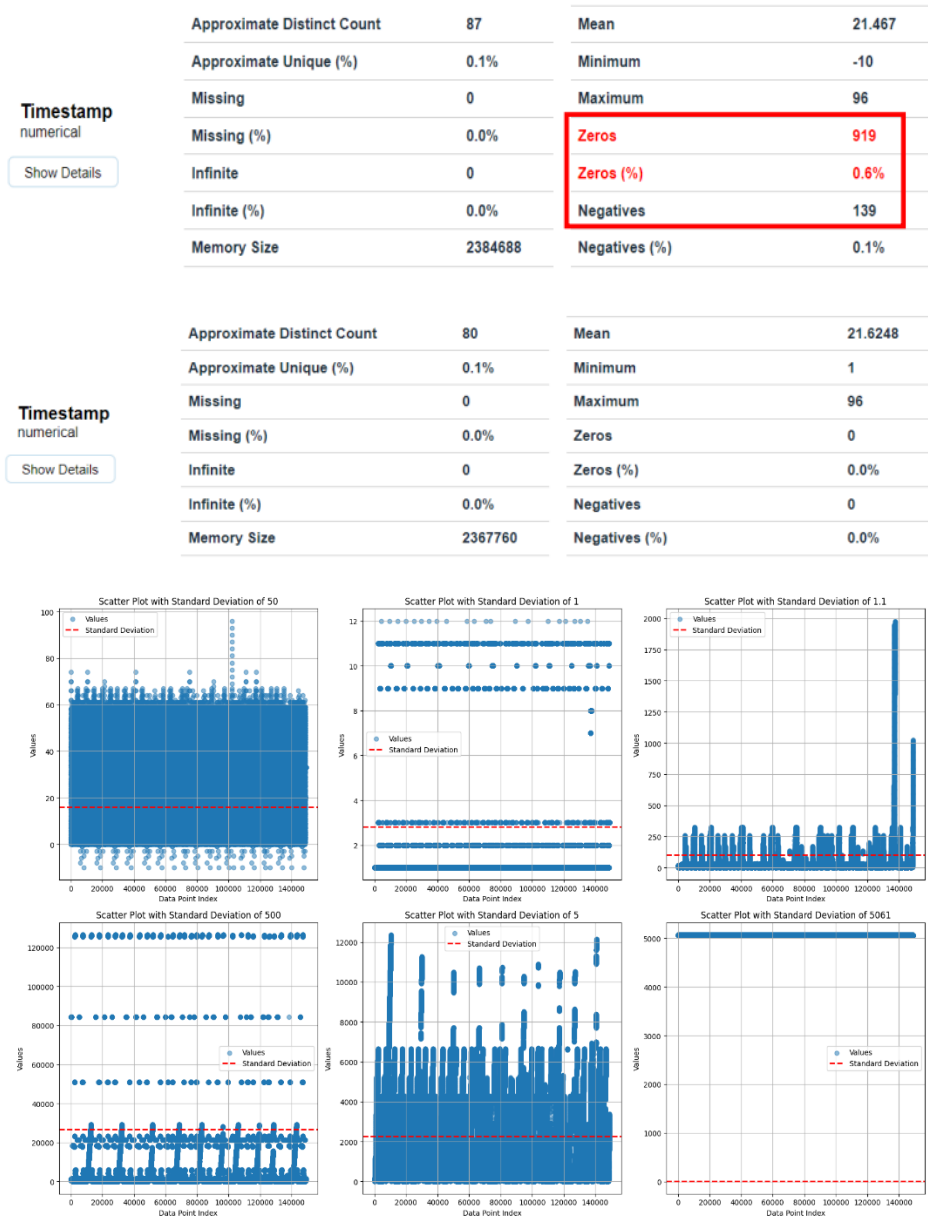


Figure 8. Anomaly in the dataset

Concurrently, an assessment of the cumulative costs associated with these ransomware types revealed that Locky, SamSam, and WannaCry still retained substantial monetary impact (Figure 10). Furthermore, an exploration of the distribution of various malware categories across ransomware types was conducted. The results indicated a relatively balanced distribution, with SSH accounting for 33.0% of instances, Spam representing 31%, and UDP scan comprising 27.6%. In contrast, NerisBonet was found to be in the minority, constituting only 8.3% of the dataset (Figure 11). Similarly, to gain a more standardised perspective and to discern the true extent of the threat posed by each ransomware variant, an analysis of the average dollars per ransomware was undertaken. This analysis yielded results divergent from the initial observation. Locky, SamSam, and WannaCry did not occupy the top three positions in this ranking. Instead, NoobCrypt, previously positioned on the lower end of the frequency spectrum, emerged as a leading contender, joined by EDA2 and DMALocker, both previously situated within the middle range (Figure 10). This result provides valuable insights into the UGRansome dataset, illustrating that while Locky, SamSam, and WannaCry may have incurred substantial cumulative damages due to their higher volume of attacks (Figure 10), they may not inflict

as much financial harm per individual attack when compared to NoobCrypt, DMALocker, and EDA2 (Figure 10). Therefore, the latter ransomware variants should be closely monitored as potential major threats, particularly if the volume of their attacks were to increase. A correlation matrix of the SAE assesses relationships between features (Figure 12), with +1 indicating a strong positive linear correlation, -1 suggesting a strong negative correlation, and values near 0 denoting weak or no correlation. It helps identify correlated features, interpret their impact on models, and guide feature selection. The strongest correlations are observed between ransomware ports and the prediction label, with a correlation coefficient of 0.27 (Figure 12). Additionally, notable correlations are found between network flow and IP address (correlation coefficient of 0.4), as well as between ransomware addresses and either USD or network flow in bytes, ranging from 0.31 to 0.38 (Figure 12). These correlations suggest significant relationships between these variables, highlighting their potential importance in understanding and predicting ransomware activity. The implications of these findings suggest that certain variables, such as ransomware ports, network flow, IP addresses, USD, and network flow in bytes, are strongly associated with ransomware activity. Understanding these relationships can aid in the development of more effective detection and mitigation strategies for ransomware attacks. By focusing on these correlated variables, organisations can enhance their cybersecurity measures to better protect their systems and data from ransomware threats. In addition, the ransomware attacks in the dataset inflicted severe financial devastation. On average, victims paid a staggering 30.69 BTC, equivalent to \$798,602 (USD) as of September 2023, with an average dollar payout of \$14,873.43 (USD). Figure 10 underscores the substantial financial toll imposed by ransomware threats. The average network traffic observed was 2021.16 bytes, with a considerable standard deviation of 2,272.54 (Figure 13).

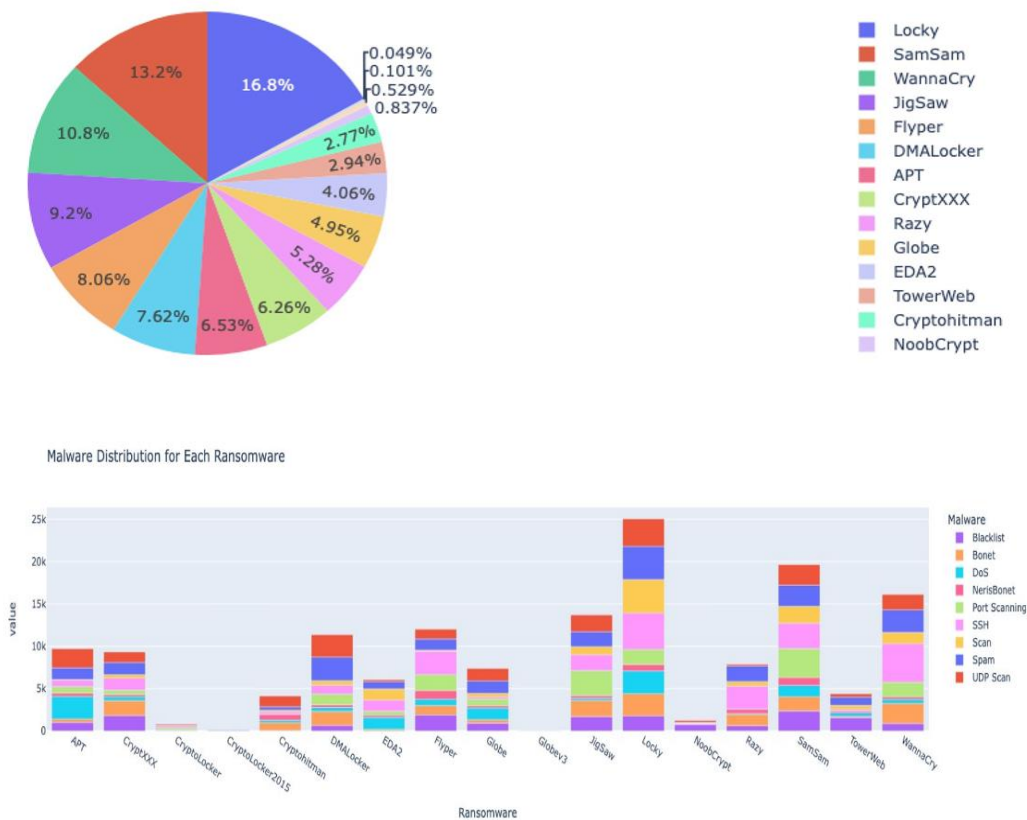


Figure 9. Distribution of attacks in the dataset

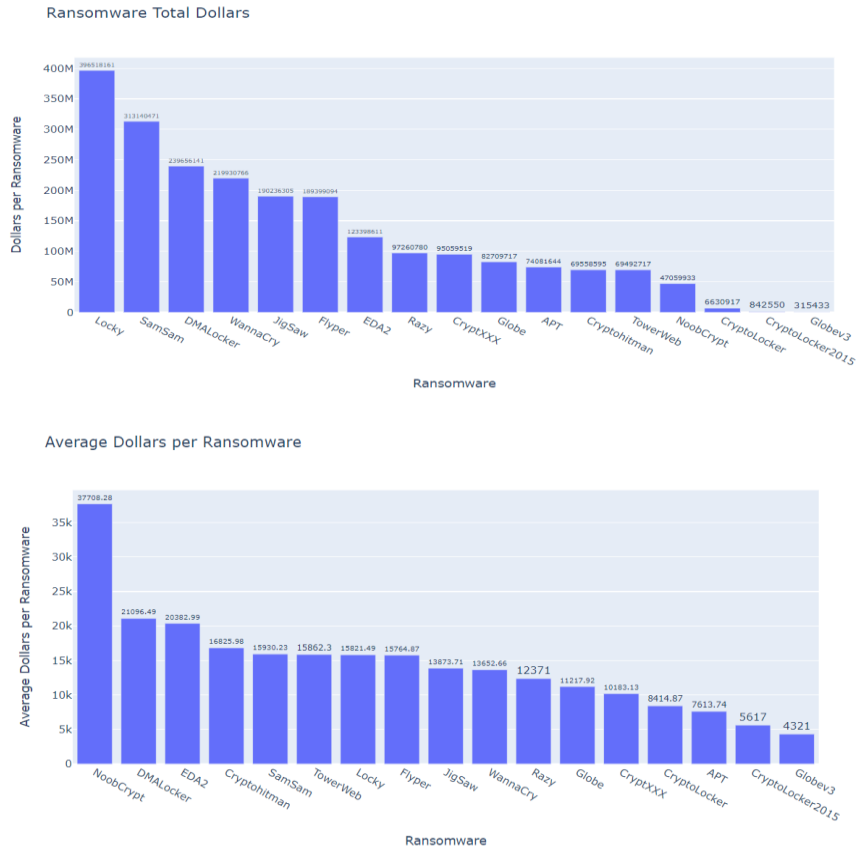


Figure 10. Financial damages of ransomware

This suggests a notable variation in values indicating spikes in network traffic triggered by zero-day threats.

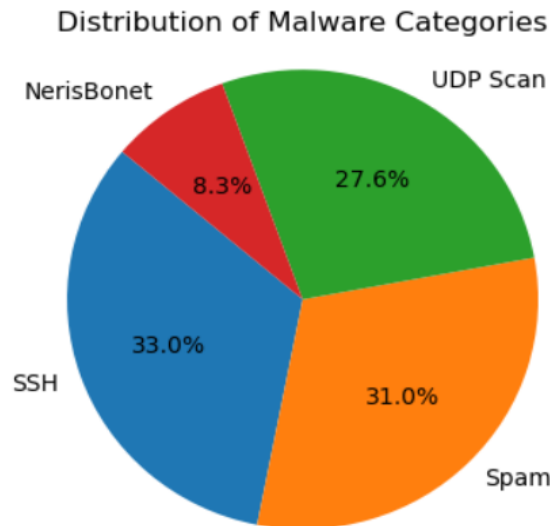


Figure 11. Malware distribution

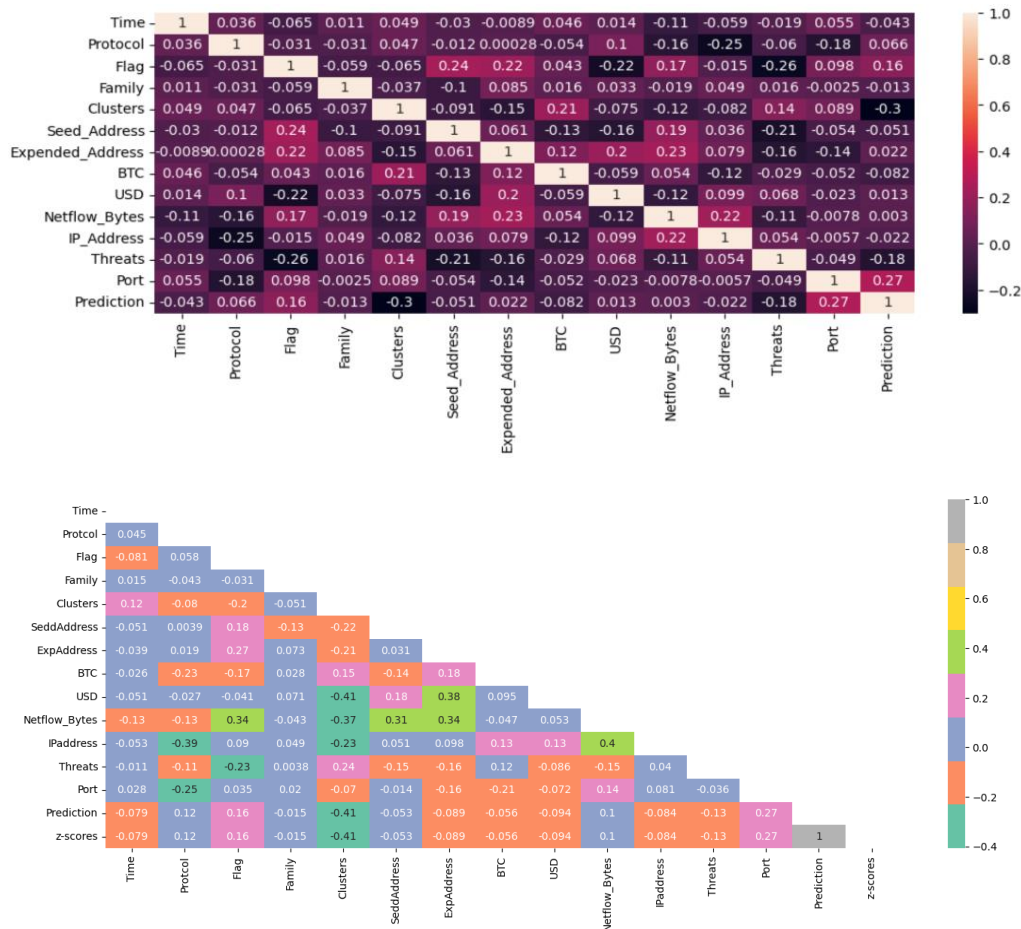


Figure 12. Correlation of features

To address this, additional feature engineering might be necessary to better balance the dataset. Figure 13 shows that *CryptoLocker* exhibited the most anomalous behaviors among ransomware. This suggests that zero-day threats like *Crypto* and *Locker*, which restrict users' access to their computers (Table 1), exhibit highly deviant behaviors compared to normal patterns due to their technical complexity and social engineering tactics [41].

4.3. SAE-LSTM Results

The LSTM model's classification results shown in Figure 14 are detailed using a confusion matrix. The matrix highlights that 17,891 instances of ransomware were correctly classified as Signature (S) types, with over 11,000 instances classified as SS and anomaly (A) types (Figure 14). This classification has an average accuracy of 98.5% (Table 5). We undertook a comprehensive comparison between the proposed SAE-LSTM and an XGBoost algorithm. The results obtained from the XGBoost, using the UGRansome dataset, have been thoughtfully summarised in Table 6. This analysis revealed the superior performance of the SAE-LSTM model over the XGBoost algorithm, which can be attributed to the effectiveness of feature selection inherent to the autoencoding approach (Figure 15, Table 5, and Table 6). The model exhibits high precision, recall, and F1 score values. This underscores its effectiveness in accurately identifying various attack types. Moreover, the balanced average scores imply that the model generalises well across different attack categories and ensures consistent performance. The confusion matrix depicted in Figure 14 provides a visual representation of the model's performance and showcases the number of TP, TN, FP, and FN predictions. The proposed model demonstrates outstanding performance in identifying signature attacks and obtained a 98.5% accuracy, showcasing its proficiency in recognising well-established ransomware patterns. This outperforms the XGBoost model, which achieved a 95.5% accuracy rate in the same task (Table 5 and Table 6).

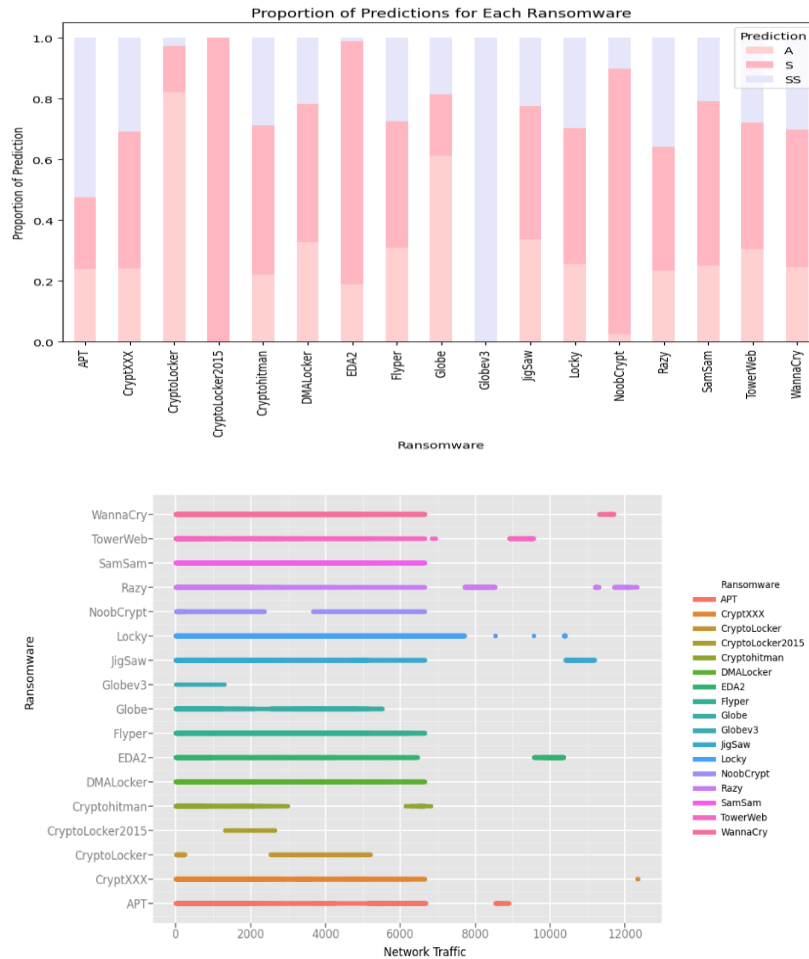


Figure 13. The 17 ransomware prediction and network traffic

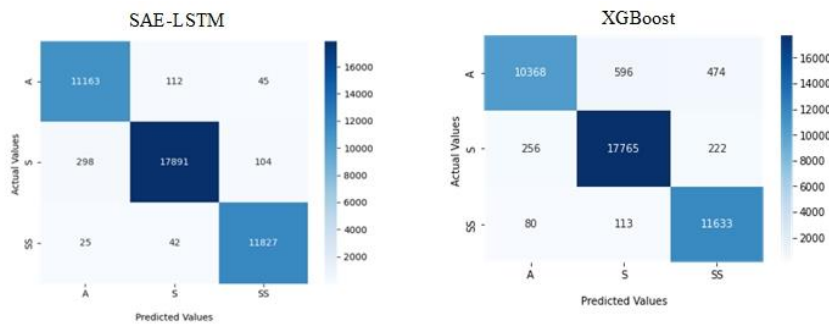


Figure 14. SAE-LSTM and XGBoost confusion matrix analysis

Figure 16 indicates that the SAE-LSTM model is more reliable in making correct predictions and has higher precision and recall compared to the XGBoost model. Therefore, SAE-LSTM is considered better for ransomware recognition. However, its slightly lower performance in identifying SS attacks highlights the challenge of detecting zero-day exploit signatures (Figure 14). Anomaly (A) attacks, representing novel threats, present a greater challenge due to their lack of discernible patterns (Figure 14). Future work in the IDS field could use the UGRansome dataset and refine the model parameters to enhance zero-day exploit detection. Table 7 presents a comparative analysis of our research findings in relation to prior studies. Although several of these studies have demonstrated commendable accuracy levels, it is essential to acknowledge certain prevalent limitations. These limitations encompass (i) the utilisation of relatively shallow learning architectures, (ii) scalability concerns, (iii) domain-specific constraints, and (iv) a heavy reliance on legacy datasets. Our research endeavors have been dedicated to surmounting these constraints through the utilisation of the UGRansome dataset and the implementation of an SAE-LSTM model, which remarkably achieved an accuracy rate of 98.5%.

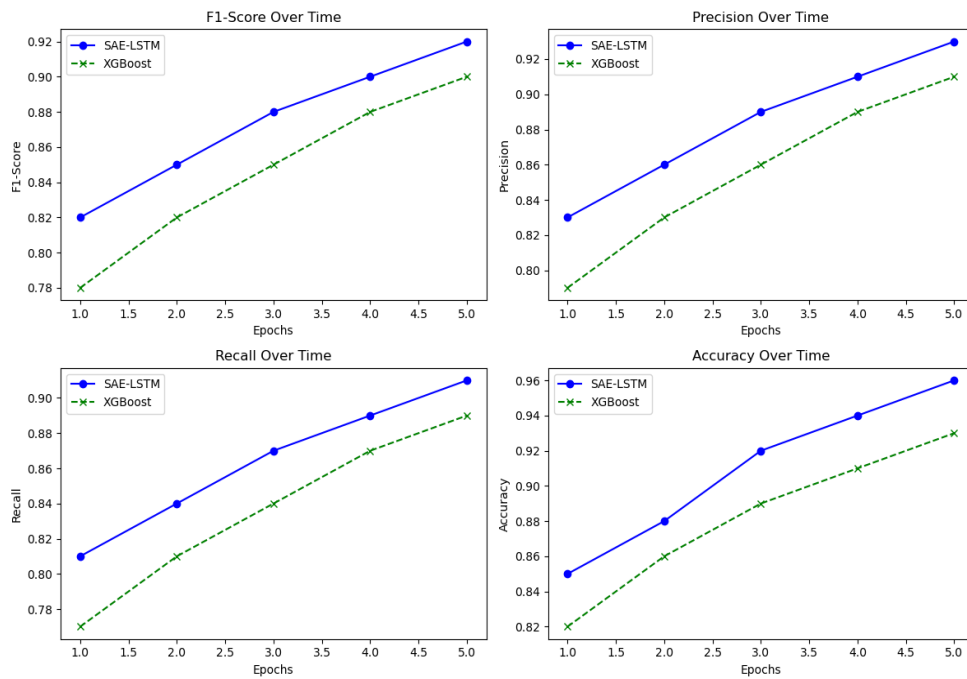


Figure 15. SAE-LSTM vs. XGBoost: Performance analysis

Table 5. SAE-LSTM evaluation

| SAE-LSTM | | | | |
|--------------------------|-----------|--------|--------------|---------|
| Prediction | Precision | Recall | F1 score | Support |
| Anomaly (A) | 97.20% | 98.6% | 97.9% | 11,320 |
| Signature (S) | 99.1% | 97.8% | 98.5% | 18,293 |
| Synthetic Signature (SS) | 98.8% | 99.4% | 99.1% | 11,894 |
| Accuracy: | | | 98.5% | 41,507 |
| Average (avg): | 98.5% | 98.5% | 98.5% | 41,507 |

Table 6. XGBoost evaluation

| XGBoost | | | | |
|--------------------------|-----------|--------|--------------|---------|
| Prediction | Precision | Recall | F1 score | Support |
| Anomaly (A) | 96.1% | 90.4% | 93.1% | 11,436 |
| Signature (S) | 95.9% | 97.3% | 96.6% | 18,249 |
| Synthetic Signature (SS) | 94.5% | 97.8% | 96.1% | 11,822 |
| Accuracy: | | | 95.5% | 41,507 |
| Average (avg): | 95.5% | 95.5% | 95.5% | 41,507 |

Table 8 displays the results of the experimental assessment conducted on various machine learning algorithms utilised in the research. The SAE-LSTM model achieved exceptional performance metrics, with 98.5% precision, 98.5% recall, and a 98.5% F1 score using 41,507 ransomware test samples (Figure 16). These results surpassed those obtained by XGBoost, RNN, LSTM, and decision tree algorithms (Figure 16). This experiment portrays the efficacy of the proposed model attributed to its SAE feature selection.

Table 8. Experimental evaluation

| Machine Learning Algorithms | Precision | Recall | F1 score |
|-----------------------------|--------------|--------------|--------------|
| XGBoost | 95.5% | 95.5% | 95.5% |
| RNN | 94% | 94% | 92% |
| LSTM | 92% | 90% | 90% |
| Decision Tree | 91% | 93% | 91% |
| SAE-LSTM | 98.5% | 98.5% | 98.5% |

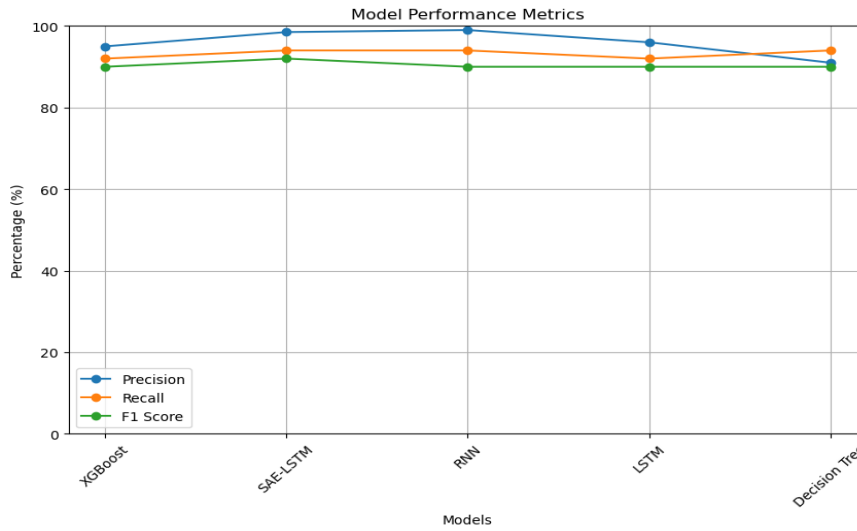


Figure 16. Performance of the SAE-LSTM model

Furthermore, Table 9 presents the evaluation of confusion matrices to analyse true and false positives. The proposed SAE-LSTM model exhibited lower misclassification rates (FP and FN) compared to other algorithms such as RNN and decision trees (Figure 17 and Table 9). The SAE-LSTM accurately classified 6,897 and 5,786 ransomware features, respectively (Table 9). These results demonstrate a substantially accurate classification rate (Figure 17 and Table 9).

Table 9. Confusion matrix evaluation

| Machine Learning Algorithms | TP | TN | FP | FN |
|-----------------------------|--------------|--------------|-----------|-----------|
| XGBoost | 4.569 | 3.897 | 63 | 54 |
| RNN | 3.766 | 2.890 | 120 | 100 |
| LSTM | 4.644 | 4.321 | 42 | 30 |
| Decision Tree | 1.886 | 700 | 200 | 145 |
| SAE-LSTM | 6.897 | 5.786 | 31 | 16 |

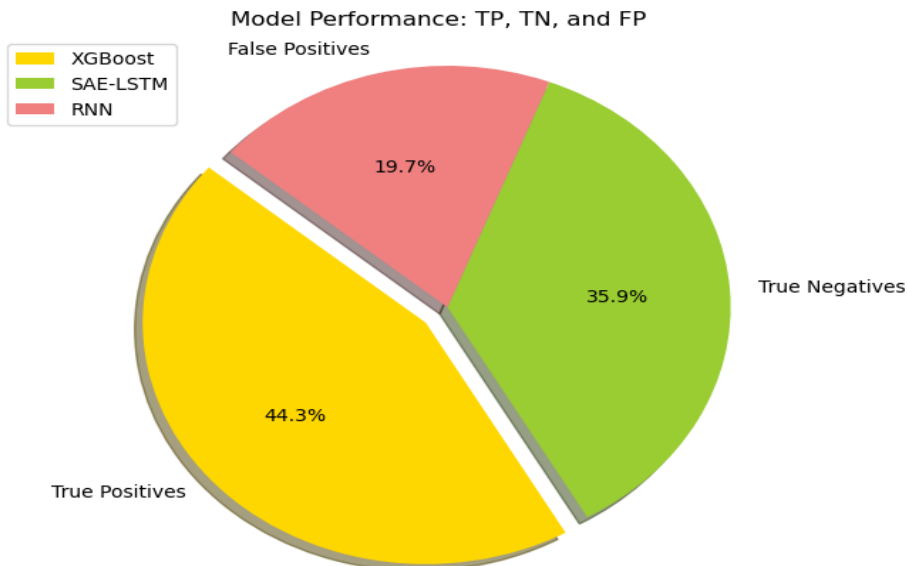


Figure 17. Accurate performance of the SAE-LSTM model

In Table 10, the performance of the SAE-LSTM model across 17 ransomware families is depicted (Figure 18). The proposed model demonstrated impressive results, achieving a 93% F1 score for CryptoLocker

attacks, a 98% precision rate for EDA2 attacks, a 99% recall rate for Locky and SamSam attacks, and a 97% precision rate for WannaCry attacks (Figure 18).

Table 10. Performance of the SAE-LSTM model for 17 ransomware families

| | Metrics | | | |
|------------------|------------|------------|------------|---------|
| | Precision | Recall | F1 score | Support |
| APT | 91% | 90% | 80% | 2,124 |
| CryptXXX | 95% | 92% | 91% | 3,012 |
| CryptoLocker | 92% | 93% | 90% | 2,569 |
| CryptoLocker2015 | 90% | 80% | 93% | 1,500 |
| Cryptohitman | 81% | 83% | 81% | 2,675 |
| DMALocker | 80% | 82.6% | 81.8% | 3,876 |
| EDA2 | 98% | 73% | 70% | 2,654 |
| Flyper | 70% | 71% | 67% | 2,232 |
| Globe | 68% | 65% | 62% | 1,432 |
| Globev3 | 63% | 60% | 59% | 3,000 |
| JigSaw | 59% | 55% | 53% | 3,500 |
| Locky | 69% | 99% | 66% | 4,000 |
| NoobCrypt | 49% | 45% | 42% | 500 |
| Razy | 39% | 49% | 40% | 800 |
| SamSam | 65% | 99% | 50% | 750 |
| TowerWeb | 55% | 50% | 49% | 950 |
| WannaCry | 97% | 52% | 41% | 1,000 |

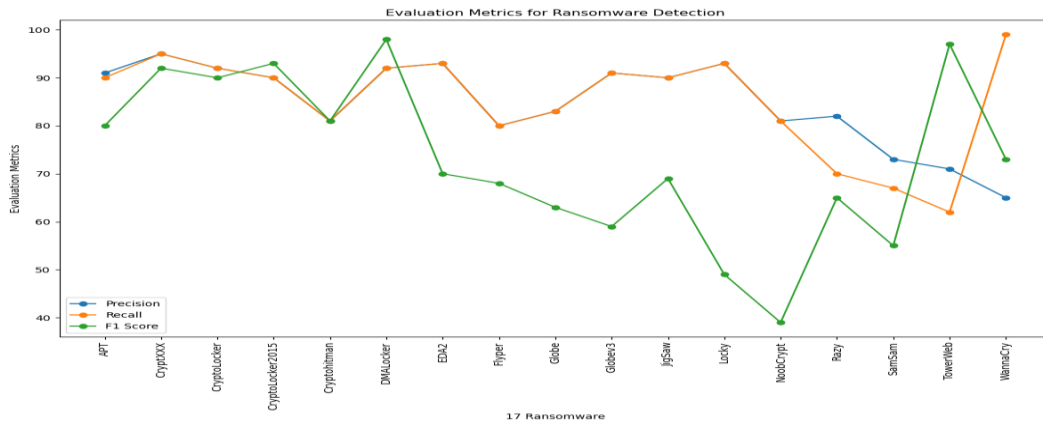


Figure 18. Performance of the SAE-LSTM using 17 ransomware families

Table 11 offers a comparative analysis of ransomware prediction accuracy using various experimental machine learning techniques. The SAE-LSTM model exhibited superior predictive capabilities, achieving a remarkable 98.5% accuracy in predicting signature, anomaly, and synthetic signature attacks (Table 11 and Figure 19). Additionally, the model successfully classified specific ransomware variants such as WannaCry, Cryptohitman, and NoobCrypt with high accuracy. These results further validate the proposed model’s effectiveness in ransomware detection.

Table 11. Accuracy of ransomware prediction using machine learning

| | S, A, and SS | | | Ransomware Family |
|---------------|--------------|--------------|---------------------|--|
| | Signature | Anomaly | Synthetic Signature | |
| XGBoost | 91% | 90% | 90% | WannaCry, SamSam, Razy, and NoobCrypt |
| RNN | 93% | 90% | 92% | Flyper, EDA2, DMALocker, and Cryptohitman |
| LSTM | 90% | 90% | 90% | CryptoLocker2015, CryptoLocker, and CryptXXX |
| Decision Tree | 97% | 91% | 93% | WannaCry, Razy, and NoobCrypt |
| SAE | 94% | 96% | 92.8% | Locky, Globe, and Cryptohitman |
| SAE-LSTM | 98.5% | 98.2% | 98.7% | Locky, JigSaw, Globev3, and Globe |

Signature (S)-Anomaly (A)-Synthetic Signature (SS) Prediction by Algorithm

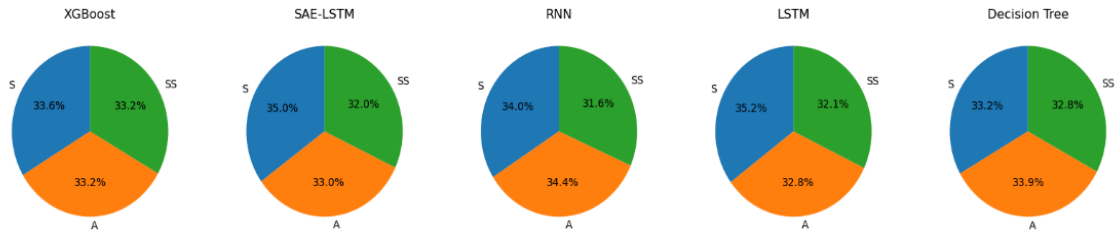


Figure 19. Predictive performance of experimental algorithms

The ability of the proposed SAE-LSTM model to accurately predict 33% of ransomware attacks as an anomaly (A) and synthetic signature (SS) is highly relevant for ransomware recognition, especially in the context of zero-day ransomware (Figure 19). Zero-day ransomware refers to previously unknown ransomware variants that exploit vulnerabilities for which no patch or defense mechanism is available. These types of ransomware can be particularly devastating as traditional signature-based detection methods may fail to recognise them. By accurately predicting ransomware attacks as anomalies or synthetic signatures, the SAE-LSTM model demonstrates its capability to identify potentially new and unknown threats, including zero-day ransomware. Predicting ransomware attacks as anomalies or synthetic signatures allows for early detection of suspicious behavior within the system. This early detection enables security teams to take proactive measures to mitigate the impact of the ransomware, such as isolating the infected system, blocking network communications, or deploying patches or updates to address vulnerabilities exploited by the ransomware. Moreover, traditional signature-based detection methods may suffer from high false-negative rates when encountering zero-day ransomware, leading to undetected infections and potential data breaches. By accurately predicting ransomware attacks as anomalies or synthetic signatures, the SAE-LSTM model can help reduce false negatives, thereby enhancing overall detection efficacy and improving the security posture of the system. In addition, zero-day ransomware poses a significant challenge to cybersecurity resilience as it exploits unknown vulnerabilities and evades traditional security measures. By accurately predicting zero-day ransomware attacks, the SAE-LSTM model contributes to enhancing cybersecurity resilience by providing early warning signals and enabling timely response and remediation actions to mitigate the impact of these threats. The presented results offer valuable insights into the effectiveness of the proposed SAE-LSTM model for ransomware detection, supporting the alternative hypothesis outlined in Section 1. The following insights provide compelling evidence of the model's efficacy and its potential to contribute significantly to the field of ransomware recognition:

4.3.1. Performance superiority

The SAE-LSTM model outperformed other machine learning algorithms such as XGBoost, RNN, LSTM, and decision tree in terms of precision, recall, and F1 score (Figure 16). This suggests that the proposed model has a higher accuracy in identifying ransomware instances, making it a promising approach for detection.

4.3.2. Feature selection impact

The use of SAE for feature selection played a significant role in improving the performance of the model in detecting ransomware (Figure 18). By selecting relevant features, the model achieved better precision and recall rates, indicating that feature selection is crucial for enhancing the effectiveness of ransomware detection algorithms.

4.3.3. Reduced misclassification rates

The SAE-LSTM model exhibited lower rates of false positives (FP) and false negatives (FN) compared to other algorithms like RNN and decision trees (Figure 19). This suggests that the proposed model has a higher level of accuracy in classifying ransomware instances correctly, thereby reducing the likelihood of misidentification.

4.3.4. Effectiveness across ransomware families

The SAE-LSTM model demonstrated consistent performance across various ransomware families, achieving high precision, recall, and F1 score rates for different types of attacks (Figure 18). This indicates the versatility and robustness of the model in detecting a wide range of ransomware variants.

4.3.5. Comprehensive prediction capabilities

The SAE-LSTM model showed promising results in predicting different types of ransomware attacks, including signature, anomaly, and synthetic signature attacks, with high accuracy (Figure 19). This suggests that the model can effectively identify diverse ransomware patterns.

4.3.6. Findings overview

In summary, the insights derived from these results highlight the effectiveness and potential of the SAE-LSTM model as a reliable tool for ransomware recognition which offers improved accuracy, reduced misclassification rates, and comprehensive prediction capabilities across various ransomware families and attack types. Looking ahead, future research avenues may focus on the limitation of our study by (i) exploring the integration of more diverse datasets, (ii) proposing an innovative model's scalability, and (iii) optimising machine learning to tackle various cybersecurity challenges. Hence, it is imperative to consider these limitations and opportunities as the field continues to evolve (Table 12). In addition, our study could not incorporate additional features such as system logs, or behavioral indicators to provide a more comprehensive view of ransomware behavior. Techniques such as distributed computing, parallel processing, or model compression were not employed to improve the model's efficiency and scalability. In addition, the application of machine learning was neglected to adaptively respond to ransomware attacks in real-time. Moreover, advanced machine learning methods such as Explainable Artificial Intelligence (XAI) could further enhance the capabilities of the proposed SAE-LSTM to detect, mitigate, and respond to evolving threats [42]. It is also crucial to address these limitations and explore new opportunities to strengthen the resilience of cybersecurity systems in the face of emerging threats. As we gaze into the future of cybersecurity research, there are exciting opportunities on the horizon. One promising avenue involves integrating a wider range of datasets encompassing various aspects of cyber threats and attacks [9, 42]. Additionally, there is room for further refinement in the scalability of our proposed model, enabling it to handle even larger and more complex datasets with ease.

Furthermore, extending the application of our approach to addressing broader cybersecurity challenges beyond ransomware detection and classification could yield invaluable insights. Therefore, as the cybersecurity landscape continues to evolve, researchers must embrace these challenges and opportunities to stay at the forefront of the field. The suggested approach in this research paper employs a comprehensive methodology for ransomware detection and classification by leveraging techniques such as data preprocessing, feature extraction using SAE, and classification with an LSTM model. This approach is compared with existing studies in Table 12 to highlight the SAE feature selection method and LSTM classifier utilised. In contrast to previous studies, which often rely on shallow learning architectures and limited feature selection methods, the suggested approach introduces a novel combination of SAE-based feature selection and LSTM classification. While some studies have achieved high accuracy rates, they may be limited by factors such as scalability issues, vulnerability in classifiers, or restricted applicability to specific datasets or scenarios (Table 12). The proposed methodology addresses these limitations by employing a robust framework that pre-processes the UGRansome dataset, extracts significant features using SAE, and utilises LSTM for accurate classification (Table 7 and Figure 20). The model achieves promising results in terms of accuracy and outperforms traditional machine learning methods (Figure 20). Furthermore, the suggested approach demonstrates adaptability and scalability, making it suitable for detecting and classifying ransomware attacks in diverse datasets and real-world scenarios. However, like any approach, there are limitations and areas for improvement, such as the need for further validation across different datasets and potential vulnerabilities in classifiers. Overall, the suggested approach represents a significant advancement in ransomware detection and classification which offers a robust and effective solution for cybersecurity practitioners (Figure 20).

The comparative Table 12 highlights several aspects in terms of ransomware recognition, including the methods used, the classifiers employed, limitations, and the achieved accuracy. The table showcases a variety of approaches used for ransomware recognition, including deep learning, autoencoders, ensemble learning, genetic algorithms, swarm optimization (SO), and more traditional methods like SVM and CNNs. Different feature selection methods are employed, such as autoencoders, Gabor transforms, and ensemble learning techniques. A range of classifiers are utilised, including deep learning-based models like RNN, LSTM networks, and CNNs, as well as traditional machine learning classifiers like XGBoost, and hidden Markov models (HMM). Nevertheless, several limitations are identified, such as scalability issues with large datasets and complex architectures, the need for labeled datasets which may be challenging to obtain, vulnerability in

classifiers leading to significant drops in true positive rates (TPR), and the necessity for further research to evaluate robustness across diverse datasets and intrusion detection scenarios. The accuracy achieved varies across different methods, with percentages ranging from 87% to 98.5%, indicating the effectiveness of the proposed approaches in detecting ransomware. Overall, the table demonstrates the diversity of approaches, the challenges faced, and the effectiveness of various methods in recognising ransomware. It underscores the importance of ongoing research and the need for innovative solutions to combat evolving ransomware threats effectively.

Table 12. Comparative analysis with existing studies

| Author | Feature Selection | Classifier | Limitation | Accuracy |
|-----------|-----------------------------|-------------------|---|----------|
| [23] | SAE | DL-API | Most methods were built on shallow learning architectures, which are not fully satisfying for malware detection problems. | 96% |
| [25] | SEAE | Ensemble | Does not consider the scalability to handle large datasets and more complex architectures. | 99% |
| [28] | Vectorization | LSTM | Designed specifically for identifying malicious JavaScript code injected into web pages. | 98% |
| [30] | Autoencoder | LSTM-SVM | This approach requires a labeled dataset for training the LSTM-SVM, which can be challenging to obtain in certain domains. | 87% |
| [18] | BAF | DNN | The study only evaluates the performance of the BAF in terms of learned features, without considering the impact on the overall performance. | 90% |
| [21] | SAE | Transforms | The performance and generalizability of the system may vary when applied to different populations or settings. | 96% |
| [19] | SAE | L21 | The study is limited to load curves and does not cover ransomware data. | 92% |
| [20] | SSAE | DL | The SSAE has only been tested on a real industrial hydrocracking process. | 92% |
| [22] | SAE | CNN | Limited data sources and the need for further exploration of other potential features. | 97% |
| [10] | SO | DL | High accuracy for zero-day attack detection using the UGRansome dataset to outperform traditional machine learning methods. | 95% |
| [24] | Gabor | DL | Vulnerability in the classifiers, as the TPR significantly dropped to zero. | 87% |
| [7] | Fuzzy | XGBoost | Further research and experimentation are required to evaluate the proposed framework's robustness across diverse datasets and its suitability for a broader range of intrusion detection scenarios. | 96% |
| [42] | RFSA | - | Necessity for the dataset expansion and testing the RFSA algorithms on another dataset. | 98% |
| [43] | - | HMM | Insufficient research on efficient intrusion detection systems in cloud environments. | 99% |
| [44] | Autoencoder | LSTM | Imbalanced issues in insider threat detection and insufficient online learning strategy which lack quality datasets due to privacy concerns. | 94% |
| [45] | - | Ensemble learning | The study relies on the Bitcoin dataset, which may not fully represent the diversity of ransomware threats in cryptocurrency transactions. | 95% |
| [46] | - | Fuzzy | Effectiveness in responding to evolving threats in real-time remains unverified. The delay in response could leave devices vulnerable to sophisticated ransomware attacks. | 88% |
| [12] | Genetic algorithm | Ensemble learning | Reliance on a single optimisation technique to enhance the classification performance of machine learning algorithms. | 89% |
| [48] | Particle swarm optimization | SVM | Requires expensive domain experts prompting the need for automated methods. | 96% |
| [49] | CNNs-BiGRU | NLP-encoder | Efficacy in detecting zero-day or polymorphic malware variants. | 98% |
| [50] | Temporal convolution | BiGRU | Effectiveness may vary across different types of ransomware attacks and system environments. | 63% |
| This work | SAE | LSTM | Restricted to ransomware detection and classification. | 98.5% |

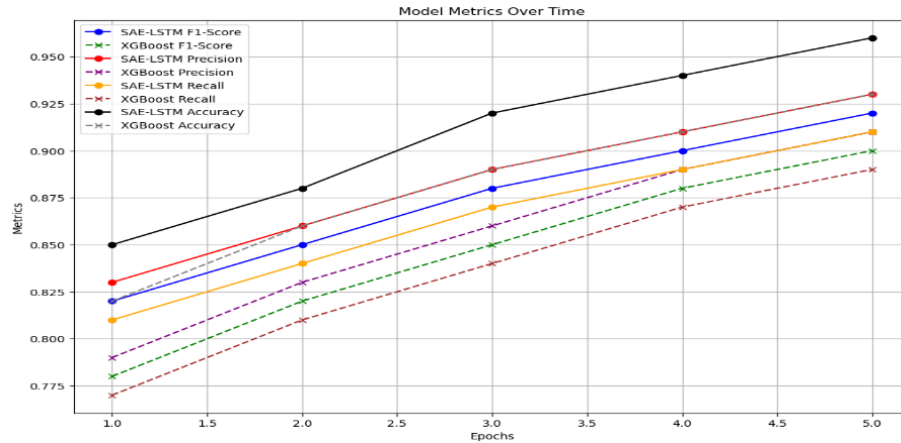


Figure 20. Reliability of the SAE-LSTM model

4.4. Limitations and Future Works

While our research has made significant strides in the realm of ransomware detection and classification, it is important to acknowledge certain limitations that may impact the applicability and generalizability of our findings:

4.4.1. Dataset limitations

Our research primarily relies on the UGRansome dataset, which may not fully capture the diversity and complexity of real-world ransomware attacks. This limitation could potentially affect the model's ability to generalise to new and unseen ransomware variants.

4.4.2. Lack of real-time adaptability

The proposed approach focuses on offline analysis and classification of ransomware threats based on historical data. As a result, it may not be well-suited for real-time detection and response to evolving ransomware attacks, which require adaptive and dynamic defense mechanisms.

4.4.3. Interpretability challenges

While our model achieves high accuracy rates in ransomware classification, the complex nature of deep learning architectures like SAE and LSTMs may hinder interpretability. Understanding the rationale behind the model's decisions and identifying actionable insights from its predictions may pose challenges for cybersecurity practitioners.

4.4.4. Generalization to other cyber threats

While our approach demonstrates efficacy in detecting and classifying ransomware threats, its applicability to other types of cyber threats, such as phishing, or insider threats, remains unexplored. Extending the model to address a broader spectrum of cyber threats would require additional research and validation.

4.4.5. Scalability concerns

The computational complexity of deep learning models like SAE and LSTMs may limit their scalability, particularly when dealing with large-scale datasets or deployment in resource-constrained environments. Addressing scalability concerns would be essential for the practical implementation of the proposed approach in real-world cybersecurity operations.

4.4.6. Ethical and privacy considerations

As with any machine learning-based approach, ethical and privacy considerations surrounding the collection, storage, and use of sensitive data must be carefully addressed. Ensuring compliance with data protection regulations and mitigating the risk of unintended consequences, such as model bias or discriminatory outcomes, is paramount. While our research represents a significant step forward in the fight against ransomware, it is imperative to recognise these limitations and areas for improvement. These challenges should be addressed in future studies to advance the field of cybersecurity in terms of developing more robust and effective defense mechanisms against evolving cyber threats. In addition to the previously mentioned future research directions, another area for exploration involves enhancing the interpretability and explainability of ransomware detection models. Future research could focus on developing methods to enhance the interpretability and explainability of ransomware detection models. Techniques such as XAI can be applied to

provide insights into the decision-making process of machine learning models using the UGRansome dataset. This will facilitate gaining a better understanding of how ransomware detection models arrive at their predictions to assist cybersecurity professionals in the interpretation of machine learning results. Considering the increasingly sophisticated nature of cyber threats, it is essential for ransomware detection models to not only accurately identify malicious activity but also provide meaningful insights into the rationale behind their decisions. Therefore, machine learning interpretability and explainability in future research endeavors can empower scientists to make more informed decisions and develop more robust defense mechanisms against ransomware and other cyber threats.

5. CONCLUSION

In the contemporary digital landscape, the pervasive menace of ransomware looms large and necessitates a proactive solution. Our research endeavors have led to the development of a novel approach aimed at effectively detecting and classifying ransomware threats. This innovative method harmoniously integrates the capabilities of an SAE for feature selection and an LSTM classifier for ransomware classification. The comprehensive process encompasses preprocessing the UGRansome dataset, employing unsupervised SAE, and fine-tuning the model through supervised learning. The culmination of these efforts has yielded a robust and adaptable model that excels in the nuanced classification of various ransomware families. Through meticulous architectural optimisations, we have achieved an exceptional accuracy rate of 98.5%, thus eclipsing the performance of traditional classifiers. This research addresses the pressing issue of ransomware detection using deep learning and paves the way for future endeavors in ransomware recognition. Future studies may explore the extension of this approach to tackle a broader spectrum of cyber threats.

ACKNOWLEDGMENTS

The authors extend their sincere gratitude to the University of Pretoria's Faculty of Engineering, Built Environment, and Information Technology for their financial support via the *UCDP Grant A1F637*.

DATASET AND CODE AVAILABILITY

<https://www.kaggle.com/dsv/7172543>

REFERENCES


- [1] C. Onwuegbuche, A. D. Jurcut, and L. Pasquale, "Enhancing ransomware classification with multi-stage feature selection and data imbalance correction," in *International Symposium on Cyber Security, Cryptology, and Machine Learning*, 2023, pp. 285–295.
- [2] N. E. Majd and T. Mazumdar, "Ransomware classification using machine learning," in *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*, 2023, pp. 1–7.
- [3] D. Krivokapic, A. Nikolic, A. Stefanovic, and M. Milosavljevic, "Financial, accounting and tax implications of ransomware attack," *Studia Iuridica Lublinensia*, vol. 32, no. 1, pp. 191–211, 2023.
- [4] V. Darwin and M. Nkongolo, "Data protection for data privacy-a south african problem?" *arXiv preprint arXiv:2306.09934*, 2023.
- [5] H. Pieterse, "The cyber threat landscape in south africa: A 10-year review," *The African Journal of Information and Communication*, vol. 28, pp. 1–21, 2021.
- [6] S. Snail ka Mtuzze and M. Musoni, "An overview of cybercrime law in south africa," *International Cybersecurity Law Review*, pp. 1–25, 2023.
- [7] M. Nkongolo and M. Tokmak, "Zero-day threats detection for critical infrastructures," in *South African Institute of Computer Scientists and Information Technologists*, A. Gerber and M. Coetzee, Eds. Cham: Springer Nature Switzerland, 2023, pp. 32–47.
- [8] A. Rege and R. Bleiman, "A free and community-driven critical infrastructure ransomware dataset," in *Proceedings of the International Conference on Cybersecurity, Situational Awareness and Social Media*, C. Onwubiko, P. Rosati, A. Rege, A. Erola, X. Bellekens, H. Hindy, and M. G. Jaatun, Eds. Singapore: Springer Nature Singapore, 2023, pp. 25–37.
- [9] M. Nkongolo, J. P. Van Deventer, and S. M. Kasongo, "Ugransome1819: A novel dataset for anomaly detection and zero-day threats," *Information*, vol. 12, no. 10, p. 405, 2021.
- [10] M. Tokmak, "Deep forest approach for zero-day attacks detection," in *Innovations and Technologies in Engineering.*, no. ISBN: 978-625-6382-83-1, 2022, pp. 45–56.
- [11] D. Shankar, G. V. S. George, J. N. J. N. S. S., and P. S. Madhuri, "Deep analysis of risks and recent trends towards network intrusion detection system," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023.

- [12] M. Nkongolo, J. P. Van Deventer, S. M. Kasongo, S. R. Zahra, and J. Kipongo, "A cloud based optimization method for zero-day threats detection using genetic algorithm and ensemble learning," *Electronics*, vol. 11, no. 11, p. 1749, 2022.
- [13] M. Nkongolo, J. P. van Deventer, and S. M. Kasongo, "The application of cyclostationary malware detection using boruta and pca," in *Computer Networks and Inventive Communication Technologies*, S. Smys, P. Lafata, R. Palanisamy, and K. A. Kamel, Eds. Singapore: Springer Nature Singapore, 2023, pp. 547–562.
- [14] A. Dairi, F. Harrou, B. Bouyeddou, S.-M. Senouci, and Y. Sun, "Semi-supervised deep learning-driven anomaly detection schemes for cyber-attack detection in smart grids," in *Power Systems Cybersecurity: Methods, Concepts, and Best Practices*. Springer, 2023, pp. 265–295.
- [15] F. Deldar and M. Abadi, "Deep learning for zero-day malware detection and classification: A survey," *ACM Computing Surveys*, 2023.
- [16] F. GUVU, I and A. SENOL, "An improved protection approach for protecting from ransomware attacks," *Journal of Data Applications*, no. 1, pp. 69–82, 2023.
- [17] A. Djenna, E. Barka, A. Benchikh, and K. Khadir, "Unmasking cybercrime with artificial-intelligence-driven cybersecurity analytics," *Sensors*, vol. 23, no. 14, p. 6302, 2023.
- [18] W. Wang, W. Y. Y. Ng, W. Li, S. Kwong, and J. Li, "Broad autoencoder features learning for pattern classification problems," in *2019 IEEE 18th International Conference on Cognitive Informatics Cognitive Computing (ICCI)**, 2019, pp. 130–135.
- [19] R. Kong, R. Lin, and H. Zou, "Feature extraction of load curve based on autoencoder network," in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, 2020, pp. 1452–1456.
- [20] Y. Wang, H. Yang, X. Yuan, Y. A. Shardt, C. Yang, and W. Gui, "Deep learning for fault-relevant feature extraction and fault classification with stacked supervised auto-encoder," *Journal of Process Control*, vol. 92, pp. 79–89, 2020.
- [21] S. Chatterjee, D. Dey, and S. Munshi, "Morphological, texture and auto-encoder based feature extraction techniques for skin disease classification," in *2019 IEEE 16th India Council International Conference (INDICON)*, 2019, pp. 1–4.
- [22] J. Kim, H. Lee, J. W. Jeon, J. M. Kim, H. U. Lee, and S. Kim, "Stacked auto-encoder based cnc tool diagnosis using discrete wavelet transform feature extraction," *Processes*, vol. 8, no. 4, 2020.
- [23] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "Dl4md: A deep learning framework for intelligent malware detection," in *Proceedings of the International Conference on Data Science (ICDATA)*. The Steering Committee of The World Congress in Computer Science, Computer, 2016, p. 61.
- [24] A. Jyothish, A. Mathew, and P. Vinod, "Effectiveness of machine learning based android malware detectors against adversarial attacks," *Cluster Computing*, pp. 1–21, 2023.
- [25] P. Panda, O. K. CU, S. Marappan, S. Ma, and D. Veasani Nandi, "Transfer learning for image-based malware detection for iot," *Sensors*, vol. 23, no. 6, p. 3253, 2023.
- [26] F. Ali, S. El-Sappagh, and D. Kwak, "Fuzzy ontology and lstm-based text mining: a transportation network monitoring system for assisting travel," *Sensors*, vol. 19, no. 2, p. 234, 2019.
- [27] M. Sewak, S. K. Sahay, and H. Rathore, "LSTM Hyper-Parameter Selection for Malware Detection: Interaction Effects and Hierarchical Selection Approach," *arXiv e-prints*, p. arXiv:2109.11500, Sep. 2021.
- [28] Y. Fang, C. Huang, L. Liu, and M. Xue, "Research on malicious javascript detection technology based on lstm," *IEEE Access*, vol. 6, pp. 59 118–59 125, 2018.
- [29] M. Nkongolo, "Fuzzification-based feature selection for enhanced website content encryption," *arXiv preprint arXiv:2306.13548*, 2023.
- [30] C. Roberts and M. Nair, "Arbitrary Discrete Sequence Anomaly Detection with Zero Boundary LSTM," *arXiv e-prints*, p. arXiv:1803.02395, Mar. 2018.
- [31] F. Suthar, N. Patel, and S. Khanna, "A signature-based botnet (emotet) detection mechanism," *Int. J. Eng. Trends Technol*, vol. 70, no. 5, pp. 185–193, 2022.
- [32] M. Komisarek, M. Pawlicki, T. Simic, D. Kavcnik, R. Kozik, and M. Chora's, "Modern netflow network dataset with labeled attacks and detection methods," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, 2023, pp. 1–8.
- [33] S. Yadav and S. Subramanian, "Detection of application layer ddos attack by feature learning using stacked autoencoder," in *2016 international conference on computational techniques in information and communication technologies (icctict)*. IEEE, 2016, pp. 361–366.
- [34] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Computer Communications*, vol. 199, pp. 113–125, 2023.
- [35] V. Nath, D. Yang, H. R. Roth, and D. Xu, "Warm start active learning with proxy labels and selection via semi-supervised fine-tuning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022, pp. 297–308.
- [36] S. A. Althubiti, E. M. Jones, and K. Roy, "Lstm for anomaly-based network intrusion detection," in *2018 28th International telecommunication networks and applications conference (ITNAC)*. IEEE, 2018, pp. 1–3.
- [37] S. A. Alsaif et al., "Machine learning-based ransomware classification of bitcoin transactions," *Applied Computational Intelligence and Soft Computing*, vol. 2023, 2023.
- [38] M. Nkongolo, "Using arima to predict the growth in the subscriber data usage," *Eng*, vol. 4, no. 1, pp. 92–120, 2023


- [39] M. Nkongolo, J. P. van Deventer, S. M. Kasongo, and W. van der Walt, "Classifying social media using deep packet inspection data," in *Inventive Communication and Computational Technologies*, G. Ranganathan, X. Fernando, and A. Rocha, Eds. Singapore: Springer Nature Singapore, 2023, pp. 543–557.
- [40] M. Nkongolo, J. P. Van Deventer, S. M. Kasongo, W. Van Der Walt, R. Kalonji, and M. Pungwe, "Network policy enforcement: An intrusion prevention approach for critical infrastructures," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, 2022, pp. 686–692.
- [41] A. Hansberry, A. Lasse, and A. Tarrh, "Cryptolocker: 2013's most malicious malware," Retrieved February, vol. 9, p. 2017.
- [42] Nkongolo Wa Nkongolo, M., "RFSA: A Ransomware Feature Selection Algorithm for Multivariate Analysis of Malware Behavior in Cryptocurrency," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 893-927, 2024.
- [43] Deep, B., & Aman, J. (2023). "Prevention and Detection of Intrusion in Cloud Using Hidden Markov Model," *International Journal of Research -GRANTHAALAYAH*, vol. 11, no. 2, pp. 40–46.
- [44] Hong, W., Yin, J., You, M., Wang, H., Cao, J., Li, J., Liu, M. and Man, C., 2023. "A graph empowered insider threat detection framework based on daily activities," *ISA transactions*, 141, pp.84-92.
- [45] Dhanya, L. and Chitra, R., 2024. "A novel autoencoder based feature independent GA optimised XGBoost classifier for IoMT malware detection," *Expert Systems with Applications*, 237, p.121618.
- [46] Mofidi, F., Hounsinou, S.G. and Bloom, G., 2024, January. "L-IDS: A Multi-Layered Approach to Ransomware Detection in IoT," in *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0387-0396. IEEE.
- [47] Mehrban, A. and Geransayeh, S.K., 2024. "Ransomware threat mitigation through network traffic analysis and machine learning techniques," *arXiv preprint arXiv:2401.15285*.
- [48] Abbasi, M.S., Al-Sahaf, H., Mansoori, M. and Welch, I., 2022. "Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection," *Applied Soft Computing*, 121, p.108744.
- [49] Maniriho, P., Mahmood, A.N. and Chowdhury, M.J.M., 2023. "API-MalDetect: Automated malware detection framework for windows based on API calls and deep learning techniques," *Journal of Network and Computer Applications*, 218, p.103704.
- [50] Jeon, J., Baek, S., Jeong, B. and Jeong, Y.S., 2023. "Early prediction of ransomware API calls behaviour based on GRU-TCN in healthcare IoT," *Connection Science*, vol. 35, no. 1, p.2233716.

BIOGRAPHIES OF AUTHORS



Mike Nkongolo Wa Nkongolo  is a Lecturer in the Department of Informatics at the University of Pretoria. He holds a PhD in Information Technology from the University of Pretoria and a Master's degree in Computer Science from the University of the Witwatersrand, Johannesburg. His research interests encompass cybersecurity, information retrieval and security, data science, machine learning, game theory, and natural language processing. He reviews for various journals, including *Automatika*, the *International Journal of Computing and Digital Systems*, the *South African Computer Journal*, and *IEEE Transactions on Education*. Dr. Mike Wa Nkongolo is a member of the South African Institute of Computer Scientists & Information Technologists (SAICSIT). He can be contacted at: mike.wankongolo@up.ac.za



Mahmut Tokmak  is an Assistant Professor at the Department of Management Information Systems, Bucak Zeliha Tolunay School of Applied Technology and Management, Burdur Mehmet Akif Ersoy University, Turkey. He holds a PhD degree in Computer Engineering. His research interests are artificial intelligence, machine learning and malware analysis. He can be contacted at: mahmuttokmak@mehmetakif.edu.tr