Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 26s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Enhanced Ransomware Threat Detection Based On AI-Powered Deep Feature Engineering Using Intellectual Cyber Swarm Intelligence Technique With Hyper Capsule Deep Neural Network

C.Porkodi[*1], Dr. P.Thiyagarajan[2]

[*1]Assistant Professor, Computer Science and Engineering, Mahendra Engineering College for Women, Kumaramangalam, Namakkal-637205

Email ID: cporkodicse@gmail.com

[2]Associate Professor, Computer Science and Engineering, Paavai Engineering College, Pachal, Namakkal-637018.

Email ID: tvm210@gmail.com

Cite this paper as: C.Porkodi, Dr. P.Thiyagarajan, (2025) Enhanced Ransomware Threat Detection Based On AI-Powered Deep Feature Engineering Using Intellectual Cyber Swarm Intelligence Technique With Hyper Capsule Deep Neural Network. *Journal of Neonatal Surgery*, 14 (26s), 592-610.

ABSTRACT

Increasing the Internet of communication in heterogeneity connects the healthcare sectors to make incredible communication, monitoring, tracking, and analyzing the patients remotely through the Internet of Things. Such data are highly protective and contain sensitive and private information. Throughout, cyber-attacks are increased by attackers without knowing the user knowledge to create data breaches with the support of various malware, applications, and viruses. The fact Ransomware virus is one of the dangerous malicious activity programs tackled by attackers to create attacks to own the healthcare industries. So, privacy and security are essential to protect PHI data records from cyberattacks. Most of the traditional methodologies failed to analysis the Ransomware behavior impacts and logical progress leads poor accuracy in detection rate. Due to inadequate feature analysis, data shifting problems lead to increasing feature dimensions, producing a lower precision rate to degrade the identification rate. To resolve this problem, we propose an Artificial Intelligence powered deep feature engineering based on Intellectual Cyber Swarm Intelligence Technique (ICSIT) with Hyper Capsule Deep Neural Network (HCDNN) to identify the Ransomware properties effectively to enhance security. Initially, the preprocessing is carried out c-score normalization to verify the actual margin of feature presence in the RaNASP dataset, and the behavioural point of RaNSAP Malware Attack Impact Rate (RMDIR) is estimated by decisive margin class predictor to marginalize the affected features. Then, Linear Support Vector Swarm Intelligence Feature selection (LSV-SIFS) is applied to select the ransomware features. Finally, the Fuzzy Inference Hyper Capsule Multi Perceptron Neural Network (FIHc-MPNN) is applied to identify the behavioral class of ransomeware attack nature. The classifier marginalizes the defect feature limits to cover the mutual dependencies of attack behavior in the classifier unit to identify the attack. The proposed system produces high performance compared to the other systems by identifying Ransomware by increasing the precession recall rate to attain a higher actual positive rate compared to other traditional methods.

Keywords: *Internet of Things, healthcare communication, Ransomware, cyber-attacks, Artificial Intelligence, feature engineering, malware detection, privacy, security.*

1. INTRODUCTION

In recent years, Ransomware has increasingly dominated the headlines of cybercrime reporting. Many attacks against organizations and individuals have caused substantial financial losses and disrupted everyday life. From basic scareware and User Interface (UI) lockers, Ransomware has developed into more complex variants, such as crypto-ransomware and, more recently, fileless and data-exfiltration ransomware. This increasing threat has led to significant research focused on analyzing and mitigating these attacks. These include identifying known Ransomware, safeguarding user files and operating systems from unauthorized changes, and preventing malware from specifically targeting victims [1]. Users are at serious risk from Ransomware, which encrypts files, locks down systems, and demands a fee to unlock them. Contemporary Ransomware, often known as crypto-ransomware, encrypts certain kinds of files on compromised computers. To obtain the decryption key, consumers must pay a ransom via specific internet payment methods [2].

Intrusion detection systems (IDS), which offer robust defense against network attackers, are crucial to network security, given the growing requirement to safeguard network resources from cyber-attacks. IDS assists in detecting and identifying abnormal activity by hackers on the computer system or network. Pre-learning features are used in machine

learning (ML) techniques to facilitate classification and prediction. IDS can be separated into supervised learning and unsupervised learning according to the training methodology. Unsupervised learning analyzes unlabeled samples to find structural information in the data set, whereas supervised learning uses labeled training samples to predict data outside the training set [3].

Moreover, IDSs have attracted significant interest in academia because they are proactive identification tools that can protect networks from internal and external attacks. An IDS is a security tool that follows firewalls and antivirus software. It replaces the network in the event of a system failure. IDSs record system network traffic and detect intrusions [4]. An IDS alerts users to intrusions at both the host and network levels. However, current IDSs face several limitations, as the dynamic nature of cyberattacks complicates detection. Additionally, large network sizes and high application volumes generate vast amounts of data, making IDS implementation challenging. Consequently, various strategies have been developed in recent years to tackle these challenges.

Ransomware constitutes a particularly malicious type of software that encrypts valuable data and subsequently demands payment from the victim in exchange for the decryption key. [5]. An effective ransomware attack can have disastrous repercussions, such as severe financial losses, reputational harm, and company disruption. As the scale and complexity of digital infrastructure increase, the risk of ransomware infection is increasing, so having an efficient detection method is becoming essential rather than optional.

Current Landscape and Challenges: Cyber-attacks on the healthcare industry have escalated, with attackers employing sophisticated techniques to breach systems and access sensitive data. Ransomware is among the most dangerous forms of these attacks, often leading to the complete shutdown of critical healthcare services and compromising patient safety. The fundamental challenge with Ransomware is its capability to encrypt sensitive data, rendering it inaccessible until a ransom is paid. Cyber attackers exploit vulnerabilities in outdated software, using various malware and viruses to infiltrate networks without user knowledge.

Traditional approaches to cybersecurity within healthcare settings have exhibited noticeable shortcomings. Many existing systems struggle to accurately detect Ransomware due to inadequate behavioral analysis, which fails to capture the evolving nature of these threats. The increasing dimensionality of feature sets often results in data-shifting problems that degrade the precision and overall identification rate of attacks. This scenario demands a sophisticated solution that not only improves detection rates but also enhances the security protocols in place.

To effectively tackle the challenges posed by Ransomware, we propose a novel system that employs an Artificial Intelligence-pivoted deep feature engineering model based on the Intellectual Cyber Swarm Intelligence Technique (ICSIT). At its core, this system integrates Hypercapsule Deep Neural Networks (HCDNN) to identify ransomware properties with heightened accuracy, thereby reinforcing security across healthcare networks.

The proposed system commences with RaNSAP data preprocessing. Utilizing c-score normalization establishes the actual margin of feature presence within the RaNASP dataset. An analysis of the RaNSAP Malware Attack Impact Rate (RMDIR) follows, using a decisive margin class predictor to determine and marginalize the features most affected by ransomware acts. This initial stage is paramount as it sets the foundation for further feature analysis and selection.

Next, we implement the Linear Support Vector Swarm Intelligence Feature Selection (LSV-SIFS) technique. This approach sifts through the comprehensive dataset to isolate the most relevant Ransomware features that contribute significantly to identifying attack patterns. By focusing on high-impact features, the system addresses the inadequacies of previous methodologies that suffered from redundant or irrelevant feature sets, thereby enhancing the detection process's overall efficiency and precision.Finally, the system employs the Fuzzy Inference Hyper Capsule Multi Perceptron Neural Network (FIHc-MPNN) to ascertain the behavioral characteristics of ransomware attacks. This classifier is particularly effective as it enhances the ability to diagnose the nature of attacks while also covering the mutual dependencies of attack behaviors, showcasing a clear advantage over traditional detection systems.

2. LITERATURE SURVEY

The author [6] discussed current trends in automatic ransomware detection and thoroughly evaluated existing methods for identifying, preventing, and mitigating ransomware attacks. However, ransomware attacks can harm computer resource owners, causing identity and privacy breaches and financial losses. Therefore, intrusion and malware detection decisions are made based on Deep Learning (DL) techniques in various contexts to prevent ransomware attacks [7]. However, as threats become increasingly sophisticated over time, detecting them is a difficult task. Besides, an effective ransomware indexing system that offers search capabilities, similarity checking, pattern classification, and clustering is proposed in order to solve the difficulties associated with deep learning techniques.

Additionally [8], by tracking and categorizing Ransomware using a common signature-based approach, similarities between various ransomware variants are found. Similarly, a new method was proposed based on static analysis to identify similarities between ransomware samples [9]. Furthermore, the proposed method extracts features from the raw bytes and significantly

improves the detection speed. The author [10] presented a method for identifying attacks with Ransomware called Cost-Sensitive Pareto Ensemble-Ransomware (CSPE-R). The suggested architecture uses a Contractive Auto Encoder (CAE) to convert the base variable feature space into a more cohesive and significant semantic feature space.

Additionally [11], error rates for the remaining data are controlled by two-stage hybrid ransomware detection models, like a Random Forest (RF) and a Markov model. However, by examining a variety of behavioral traits, ransomware detection methods build complex models. By choosing ransomware detection methods that use encryption algorithms and various file formats, the complexity of the detection can be reduced. Changes in entropy, however [12], cannot be used to identify ransomware-infected files. The study [13] described methods to determine the generation of these encrypted files, regardless of the type of Ransomware. Eight distinct sets of files were produced as a result of many well-known ransomware assaults in the real world, such as Phobos, SotinoKibi, and NetWalker. A presented [14] machine learning (ML) model can detect ransomware behaviors by analyzing system irregularities. Additionally, the suggested ML technique offers an effective method to integrate with intrusion detection systems (IDS) and find zero-day ransomware attacks in data. The Deep Convolutional Generative Adversarial Networks (DCGAN) generator that the author [15] demonstrated performed effectively was transformed into a GAN dispatching generator.

Table 1. Ransomware IDS Detection Based on Deep Learning

Author	Year	Technique Used	Dataset	Performance Evaluation	Limitation	Accuracy
Y. Yang [16]	2020	Deep Neural Network (DNN)	UNSW-NB15	F1 score and false positive rate.	Network security problems continue to occur due to the Internet's rapid development, and data security presents difficulties.	89.36%
Sai Chaitanya Kumar, [17]	2024	Deep Residual Convolutional neural network (DCRNN)	Cicddos2019 and CIC-IDS-2017	Processing time.	These IDS are unable to identify zero-day attacks.	89.4%
Arif, J. M [18]	2021	RF	Androzoo dataset	True Positive Rate	A thorough security evaluation is essential to identify RaNSAP malware.	91.6%
Revathy, G [19]	2021	Gradient Boosting Classifier	UNSW-NB15	False negative rate, accuracy, F-score	IDS technology is essential to detection and prevention systems that eliminate issues brought on by hackers in network settings.	93%
Rajest, S S [20]	2024	CatBoost Classifier	UNSW-NB15	Accuracy, Recall	These issues also affect initiatives that address the digital divide and encourage online protection.	92.53%
Urooj [21]	2023	Weighted GAN (wGANs)	Pre-encryption dataset c	Accuracy, Precision	Due to the low amount of data collected prior to the ransomware assault, the assumptions are invalid.	0.9533

Zanoramy [22]	2024	Naive Bays (NB)	RF and LR	Accuracy	Developing dynamics is crucial for cybercrime specialists in the digital underworld.	0.951

The paper [23] offered ML methods such as RF and Extreme Gradient Boosting (XGBoost) for performance evaluation. The findings demonstrate that the XGBoost model for feature selection may attain unmatched threat identification accuracy. The method improves conventional algorithms that depend on static signatures or scant behavioral analysis in terms of detection accuracy and false alarm rates [24]. Effective feature selection methods, such as chi-square feature selection and informative feature extraction [25], have been presented to extract the essential features in order to enhance IDS's detection capabilities.

3. PROPOSED METHODOLOGY

The advent of the Internet of Things (IoT) has revolutionized communication across various sectors, especially in healthcare. IoT facilitates incredible communication, remote monitoring, and tracking of patients, thereby enhancing efficiency and promoting patient-centered care. However, this proliferation of interconnected devices and systems raises significant cybersecurity concerns. Healthcare organizations increasingly become targets for cyber attackers, primarily as they handle sensitive and protected health information (PHI). Among the excess of malicious cyber activities, ransomware attacks stand out as particularly destructive, posing critical threats to healthcare integrity and patient privacy. Traditional methodologies for ransomware detection have shown limited efficacy due to their inability to effectively analyze the complex behaviors of Ransomware, leading to often poor detection rates.

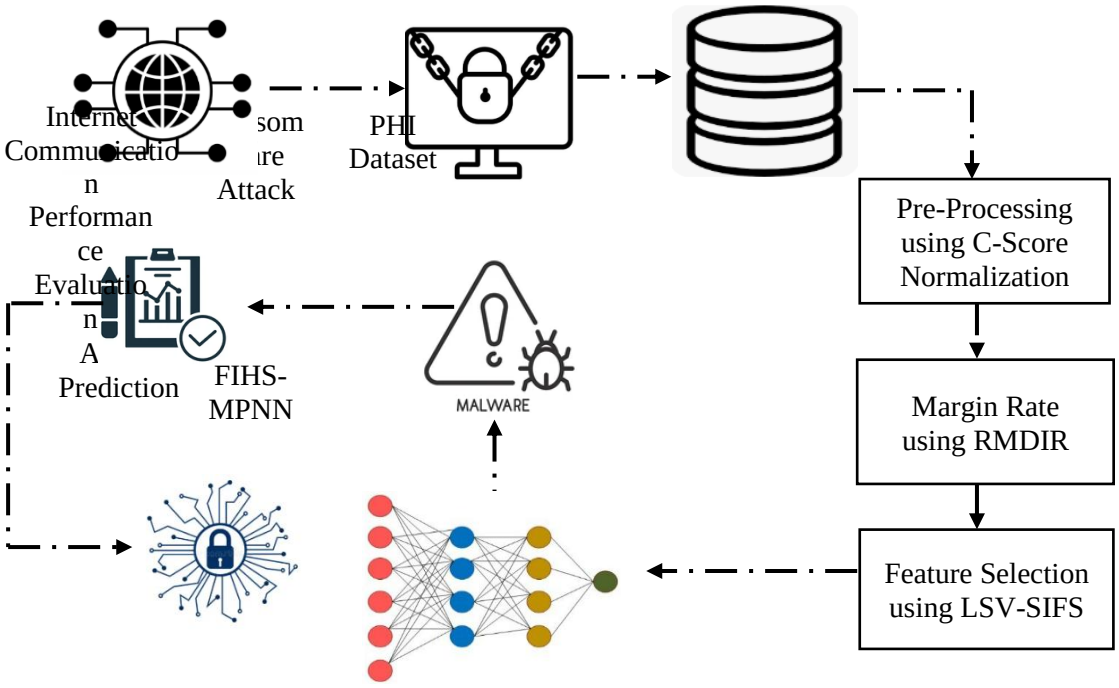

Figure 1 proposed workflow architecture ICSSIT-HCDNN

This paper proposes an advanced Artificial Intelligence (AI) powered system designed to improve the identification and monitoring of ransomware threats in the healthcare sector. Figure 1 shows the proposed workflow architecture ICSSIT-HCDNN. To address these challenges and enhance the security framework within the healthcare sector, we propose a multi-faceted system that integrates Artificial Intelligence, Intellectual Cyber Swarm Intelligence Technique (ICSIT), and Hyper Capsule Deep Neural Network (HCDNN). This innovative system aims to identify ransomware properties effectively and provide enhanced security for sensitive health data.

A) Dataset description

Ransomware addresses were widely taken from the three accepted studies: Montreal, Princeton, and Padua. Six attributes of an address (income, neighborhood, weight, length, count, loop) are extracted from a heterogeneous RaDSAP data log from network at each 24-hour snapshot. In 24 ransomware families, at least one address appears over a 24-hour time frame. Crypto Locker has 13 addresses, each of which occurs over 100 times. Four addresses have ransomware labels between the conflicting Montreal and Padua datasets. The ransomware families APT (Montreal) and Jigsaw (Padua), respectively, have

a single P2SH address (starting with "3"). And other addresses are regular addresses beginning with "1".

There are 10 features defined in this dataset. They are string, bitcoinaddress as agent, years as integers (first day as 1 and last day as 365), length as integer, weight as float, count as integer, looped as integer as label type string ransomware family name or white i.e. unknown Ransomware.

Table 1: ransomware dataset

	address	year	day	length	weight	count	looped	neighbors	income	label
2	111K8kZAl	2017	11	18	0.008333	1	0	2	1E+08	princetonCerber
3	1123pJv8j:	2016	132	44	0.000244	1	0	1	1.00E+08	princetonLocky
4	112536im7	2016	246	0	1	1	0	2	2.00E+08	princetonCerber
5	1126eDRw	2016	322	72	0.003906	1	0	2	71200000	princetonCerber
6	1129TSjKt:	2016	238	144	0.072848	456	0	1	2.00E+08	princetonLocky
7	112AmFA7	2016	96	144	0.084614	2821	0	1	5.00E+07	princetonLocky
8	112E91jxS	2016	225	142	0.002089	881	0	2	1.00E+08	princetonCerber
9	112eFykaD	2016	324	78	0.003906	1	0	2	1.01E+08	princetonCerber
10	112FTiRdJ	2016	298	144	2.302828	4220	0	2	8.00E+07	princetonCerber
11	112GocBgl	2016	62	112	3.73E-09	1	0	1	5.00E+07	princetonLocky
12	112gXL4Ae	2013	317	4	0.007143	2	0	1	1.00E+08	montrealCryptoLocke
13	112nEBUa	2016	247	0	1	1	0	2	1.09E+08	princetonCerber
14	112Ns49U	2016	146	144	0.877485	4817	0	1	1.04E+08	montrealCryptXXX
15	112vq2Wt	2017	3	4	0.015625	1	0	2	5.60E+07	princetonCerber
16	112wED5u	2016	158	56	3.05E-05	1	0	1	1.20E+08	montrealCryptXXX
17	112wED5u	2016	156	8	0.75	2	0	4	2.40E+08	montrealCryptXXX
18	112wjYgW	2016	273	144	0.008747	1168	0	1	5.50E+08	princetonLocky

The above table 1 is refer that the elements of the dataset related to the ransomware detection. In that the 10 different elements are available in the table.

B) Preprocessing C- score normalization

The first step in our proposed system involves preprocessing the data using c-score normalization. This technique ensures that features are standardized and verified for their relevance in the context of the RaNASP dataset. C-Score normalization is a basic data preprocessing method in statistics that is mainly used to bring a set of scores to the same parameter range for easier comparison as well as superior performance in a model. In cases where the variables in two datasets have different scales or variances, it is favorable. Standardization locates the data by making it mid-range, thus giving transformed data a mean closer to or equal to zero. Of those, it normalizes it to the range of -1 to 1 since it enhances the comparability of the features. It depends upon the range of the given data and minimizes the impact of outliers. In equation 1, we perform C-Score normalization formulae,

$$C = \frac{i-\mu}{d} \quad (1)$$

Let's assume i is individual data, μ is a mean of all values in the dataset, and d is the range of the data. This equation shifts the data so that the new distribution is centered around zero. In equation 2, we compute the μ,

$$\mu = \frac{\sum_{x=1}^{N} i_x}{N} \quad (2)$$

Let's assume x is a data point and N is the total number of data points. This equation centers the data in normalization by subtracting it from each data point. Then, we compute the range of data through equation 3, which evaluates the difference among the minimum and maximum values in the dataset.

$$d = \mathrm{Max}(I) - \mathrm{Min}(I) \quad (3)$$

Then we compute the variance through equation 4 to spread or dispersion of data points around the μ,

$$\sigma^2 = \frac{\sum_{x=1}^{N}(i_x-\mu)^2}{N} \quad (4)$$

This equation can illuminate the data spread in the set, but it is not utilized in C-Score normalization except when followed by further assessment. So, we perform standard deviation σ through equation 5.

$$\sigma = \sqrt{\sigma^2} \quad (5)$$

In this equation, it is an alternative to d for scaling if variability around the μ is of greater interest. Then, we perform Z-Score

normalization through equation 6,

$$Z = \frac{i-\mu}{\sigma} \quad (6)$$

This equation is used commonly for statistical analysis. By the following, we min-max normalization through equation 7, to rescale data into specific ranges as $[0,1]$ or $[-1,1]$,

$$i' = p + \frac{(i-\text{Min}(I))(q-p)}{\text{Max}(I)-\text{Min}(I)} \quad (7)$$

Let's assume p as the desired upper bounds and q as the desired lower bounds. This equation ensures all values are within a predefined range while preserving relative differences between data points. Then, we adjust the weight based on each data point through Equation 8,

$$\mu_u = \frac{\sum_{x=1}^{N} u_x i_x}{\sum_{x=1}^{N} u_x} \quad (8)$$

Let's assume u is weight, and μu is weighted mean. This equation incorporates the importance or significance of specific data points when centering data for normalization. Then we compute the C-Score with μ_u through equation 9 to allow different scaling factors F,

$$C = \frac{i-\mu_u}{F} \quad (9)$$

Thus, this equation can be more focused on particular data values and select scaling factors according to the character of the set. In C-Score normalization, it is clearly stated that the features under consideration have different scales or units, so normalizing the data brings other qualities to the same level, thereby enhancing the probability of convergence of the models as well as the accuracy of the final model that is produced. They make it easier to compare data since each data point is presented relative to the data mean and variability range. The method can also be generalized so that weighted means or other coefficients are used instead of coefficients, respectively. Combined with suitable scaling measures, it is equally potent in minimizing the effects of outliers in datasets. It helps to keep the relative position of the points depending on the meaningful value relationships. These features can be used with various datasets and, at the same time, improve the results obtained from the analysis.

C) RaNSAP Malware Attack Impact Rate (RMDIR)

By assessing the actual margins of feature presence, we can accurately estimate the RaNSAP Malware Attack Impact Rate (RMDIR) through a decisive margin class predictor. This process will help marginalize the affected features and establish a robust baseline for further analysis. The importance of features in decision trees is determined by the extent to which they decrease randomness in a given set (e.g., based on the Gini Index or Information Gain). The scores nearer to the root node are more significant because these features have the best separation of the data set. The method offers interpretability and visualization of the importance of features in classification or regression problems. It can handle datasets with large numbers of features, which is typical of transmission datasets. In equations 10 to 13, we perform the impurity measures; they have three standard measures: Gini Index D, Entropy $\mathbb{E}$, and Variance Reduction. In equation 10, we perform the Gini Index D,

$$D = -\sum_{x=1}^{n} q_x^2 \quad (10)$$

Let's assume x is a class, q is a proportion of samples, and n is a number of classes. This equation is used to measure the likelihood of incorrect classification at a node. Then we perform Entropy $\mathbb{E}$ in equation 11,

$$\mathbb{E} = -\sum_{x=1}^{n} q_x \, log_2(q_x) \quad (11)$$

In this equation, we quantify the information needed to classify a sample. After that, we perform variance reduction for regression tasks through equation 12,

$$\Delta\sigma^2 = \sigma_p^2 - \left(\frac{N_L}{N}\sigma_l^2 + \frac{N_R}{N}\sigma_r^2\right) \quad (12)$$

Let's assume σ^2 is the variance of target values, p is the parent node, N is the total number of samples, and $\left(\frac{N_L}{N}\sigma_l^2 + \frac{N_R}{N}\sigma_r^2\right)$ is the number of samples in the left and right child nodes. This equation is used for continuous targets. These equations evaluate the "purity" of a node after a split. By following, we perform Information Gain G through equation 13,

$$G = \mathbb{E}(p) - \left(\frac{N_L}{N}\mathbb{E}(l) + \frac{N_R}{N}\mathbb{E}(r)\right) \quad (13)$$

Let's assume E is Entropy, $N(p)$ is the total number of samples in the parent node, and $\left(\frac{N_L}{N}\mathbb{E}(l) + \frac{N_R}{N}\mathbb{E}(r)\right)$ is the number of samples in the left and right child nodes. This equation is used to measure the reduction in entropy after a split. Then we perform the Gain Ration H through equations 14 and 15,

$$H = \frac{G}{S} \quad (14)$$

Here, S is known for Split Information, then compute in equation 6,

$$S = -\sum_{x=1}^{m} \frac{N_x}{N} \log_2 \left(\frac{N_x}{N}\right) \quad (15)$$

Here, the m is represented as the number of partitions. These equations are used to counteract the bias toward features with more levels. After counteracting the bias, we perform the feature importance F through equation 16,

$$F(f) = \sum_{k \in K} \Delta X_k . \frac{N_k}{N} \quad (16)$$

Let's assume f is a feature, K is a set of all nodes, I is impurity, k is node, and N is a total number of samples. This equation is computed as the average reduction in impurity due to splits on a feature. When we apply the following stopping criteria to prevent overfitting: max depth and min samples from the leaf, DT can still reduce dimensionality, computational time, and even the quality of the model by selecting the essential characteristics. When used with appropriate regularization techniques (e.g., restricting the depth of trees or using ensemble methods), they remain an efficient and accurate tool for feature selection in RaDSAP and other analyses.

D) Linear Support Vector Swarm Intelligence Feature Selection

Once the features have been normalized, we employ Linear Support Vector Swarm Intelligence Feature Selection (LSV-SIFS) to identify the most relevant features for ransomware detection. This unique hybrid approach combines the strengths of Support Vector Machines (SVM) with swarm intelligence algorithms to optimize feature selection. By minimizing redundancy and focusing on key features, this method enhances the system's capacity to make accurate predictions about potential ransomware attacks.

As observed, LR is less computational and ideal for small to moderately sized datasets. It is used to eliminate many features, enhance the model, and minimize overfitting if present. In equation 17 we perform Linear Regression to establish a relation among malware features $i_1, i_2, \ldots, i_n$ and target prediction j,

$$j = \beta_0 + \beta_1 i_1 + \beta_2 i_2 + \cdots + \beta_n i_n + \epsilon \quad (17)$$

Let's assume β_0 as the initial starting value of j when whole $i_x = 0$, $\beta_1, \beta_2, \ldots \beta_n$ to evaluate how much each feature affects the target. This equation arranges the foundation of the LR model. By following we predict the output values on the basis of current values of the features $i_1, i_2, \ldots, i_n$ through equation 18,

$$\hat{j} = \beta_0 + \beta_1 i_1 + \beta_2 i_2 + \cdots + \beta_n i_n \quad (18)$$

Here, $\hat{j}$ for predicted output, the equation evaluates $\hat{j}$ by plugging in the values of the features and the estimated coefficients, and it is used to make predictions for new data after training the model. Then we measure error for how the LR predicts the actual values j_x by equation 19,

$$Y(\beta) = \frac{1}{m} \sum_{x=1}^{m} (j_x - \hat{j}_x)^2 \quad (19)$$

Here, x is represent for data point, in this equation we square the differences to ensure they are positive and sum them up to calculate the overall error. This equation is used to minimize the error during the training process. After we estimate the optimal coefficients to reduce the error $Y(\beta)$ through equation 20,

$$\beta = (I^T I)^{-1} I^T J \quad (20)$$

Here, the X is used to develop a design matrix where rows illustrate data points, and columns illustrate features, J as the vector of actual target values, and β for best fitting weights. This equation uses matrix operations to find the coefficients efficiently. Then, we compute how the LR explicates the variation in the target variable j through equation 21,

$$R^2 = 1 - \frac{SS_r}{SS_t} \quad (21)$$

Let's assume R2 evaluating the value among 0 and 1; let 1 for perfect prediction, 0 for no predictive result, SS_r as residual error, and SS_t as total variance in j. While interpreting the coefficients with large absolute values, it was found that they portend a high predictive value of the covariate in question. It means that the user should strip features with coefficients close to zero or high p-values (insignificant values). This reduces the dimensionality, reducing the models to only those that will be most needed for the prediction process. LR also offers a structured approach to screen and select the salient features from malware data for feeding into the modeling process, thus achieving effectiveness and interpretability while offering reasonable accurate results with reasonable complexity.

One of the approaches to feature selection using SVM is building by using the property of SVM to find the best hyperplane that can be separated into classes in the feature space. This one quantifies and ranks features based on how they contribute to the decision boundary. Some of the features are given more weight than others, thus making it easier to settle for the best

features to refine the model of significance upon which to expand. SVM finds the maximum margin separative hyperplane in the feature space. We learned previously that the weights of the SVM model could be derived from the decision function and that these represent feature importance. It also needs to be mentioned that more significant weight values mean higher significance in the classification process. SVM enables the selection of only one feature from each cluster and thus decreases the number of features in the set, making calculations quicker and improving model generalization. SVM has constraints regarding the linearity of separability, and it has the potential to convert features into higher dimensions using kernel functions like RBF, polynomial, etc. It cuts the processing time down by excluding variables that are unnecessary or unhelpful. In equation 22, we perform the SVM decision boundary,

$$u.i + b = 0 \quad (22)$$

Let assume, u as weight vector, it is used to regulates the orientation of the hyperplane, i as feature vector of data points, and b as bias term formalized as hyperplane for vectors.

$$\min_{u,b} \frac{1}{2}\|u\|^2 \quad \text{Focus to,} \quad j_x(u.i_x + b) \geq 1, \forall x \quad (23)$$

Let's assume x is a sample, j is a class label, and $\|u\|^2$ is the regularization term to preclude overfitting. This equation certifies that entirely points are appropriately classified through a margin of at least 1. After classifying the entire points with appropriate, we compute the feature importance F through equation 24. The u affords insights into the importance of features y,

$$F = |u_y| \quad (24)$$

Here, significance of a feature increases when it has a high absolute value of u_y meaning that it is highly influential in defining the hyperplane. Then, we separate the non-linear dataset by using kernel trick through equation 25. For non-linear data sets, SVM transforms the actual data points into a higher dimension by use of a kernel function K.

$$K(i_x, i_y) = \emptyset(i_x).\emptyset(i_y) \quad (25)$$

Let's assume $\emptyset(i)$ as a mapping function to a higher-dimensional space, and $K(i_x, i_y)$ as a Kernel function like Radial Basis Function (RBF), linear, and polynomial. This equation permits SVM to evaluate feature importance indirectly in the transformed space. After assessing the feature importance, we perform dual formulation through 26 and 27,

$$\max_{\alpha} \sum_{x=1}^{n} \alpha_x - \frac{1}{2}\sum_{x=1}^{n}\sum_{y=1}^{n} \alpha_x \alpha_y j_x j_y K(i_x, i_y) \quad (26)$$

Focus to,

$$0 \leq \alpha_x \leq P, \sum_{x=1}^{n} \alpha_x j_x = 0 \quad (27)$$

Let's us assume α_x is the Lagrange multiplier and P is a regularization parameter. This equation is used to control trade-offs among margin width and classification errors, and it is used to simplify the evaluations. Thus, SVM provides quantification of the u obtained during the optimization process to determine the relative measure of importance of each feature used in the construction of the boundary of separation. This also assists in choosing a number of features that cause a significant change in classification accuracy while leaving out unimportant ones. It has been identified that feature selection by SVM achieves dimensionality reduction of the dataset, thus faster computational time and efficient models. SVM also improves the classification accuracy and generalization ability of unknown data by computing the most important features. In this respect, as capable of employing both linear and non-linear kernels, SVM remains a versatile tool capable of application in many fields with particular reference to imaging. The margin maximization principle makes it possible to protect the models against overfitting regardless of the data's dimensionality. Despite the fact that the SVM is a powerful tool for image classification, it is equally important to note that it is also a good tool for feature selection, especially if the number of selected features has to be meaningful for an improvement in the performance of the classifier as well as a reduction in the computational burden in applications.

E) Fuzzy Inference Hyper Capsule Multi Perceptron Neural Network

The crux of our proposed system lies in the Fuzzy Inference Hyper Capsule Multi Perceptron Neural Network (FIHc-MPNN). This sophisticated deep learning model is designed to identify the behavioral class of ransomware attacks effectively. By analyzing the relationships between the selected features, the FIHc-MPNN can marginalize defect feature limits and accommodate mutual dependencies in attack behavior. This enables it to recognize complex patterns that signify ransomware presence. Initialize the fuzzy inference in the membership function (W) as

$$\sum_{y=1}^{d} w_{xy} = 1, \ \forall x \in \{1,2,\dots,V\} \quad (28)$$

here, V actual feature limits margin, and d points affected feature limits

$K(W, d_1, d_2, \ldots, d_d) = \sum_{x=1}^{d} K_x = \sum_{x=1}^{d} \sum_{y=1}^{v} w_{xy}^{p} g_{xy}^{2}$ (29)

w_{xy} is 0 to 1 and D_x is the centroid of cluster X. Did is the Euclidean distance between the $x-th$ centroid and the $y-th$ data point. $p \epsilon [1, \infty]$ is a weighting function;

Identify the mean active mid-margin of the member set using the following equation,

$$z_y = \frac{\sum_{x=1}^{V} w_{xy}^{p}.a_x}{\sum_{x=1}^{V} w_{xy}^{p}} \quad (30)$$

here, w_{xy} is stand for membership degree of feature points x in cluster y, a_x is stands for feature points value, p stands for fuzziness index, characteristically $(p = 2)$.

Estimate the convergence vector space to correlated feature margins.

$$\left\| W^{(h+1)} - W^{(h)} \right\| < \varepsilon \quad (31)$$

$$\left\| Z^{(h+1)} - Z^{(h)} \right\| < \varepsilon$$

here, the membership $W^{(h+1)}$ and $W^{(h)}$ are the matrices of $h+1$ and h respectively, the clusters centers $Z^{(h+1)}$ and $Z^{(h)}$.

Divide the data points from vector space,

$$w_{xy} = \frac{1}{\sum_{q=1}^{d} \left(\frac{g_{xy}}{g_{qy}}\right)^{2/(p-1)}} \quad (32)$$

$$d_{xy} = \frac{\sum_{y=1}^{v} w_{xy}^{p} a_y}{\sum_{y=1}^{v} w_{xy}^{p}} \quad (33)$$

Stop iteration when $max_{xy} \left\{ \left| w_{xy}^{(q+1)} - w_{xy}^{(q)} \right| \right\} < \varepsilon$. where ε is the active scalar margin 0 and 1, and q is the iteration step size.

Step 1: Set $W = [w_{xy}]$ matrix, $W^{(0)}$

Step 2: At q-phase: calculate the midpoints vectors $D^{(q)} = [d_y]\ through\ W^{(q)}$

$$D_y = \frac{\sum_{x=1}^{V} w_{xy}^{p} A_x}{\sum_{x=1}^{V} w_{xy}^{p}}$$

Step 3: Update $W^{(q)}, W^{(q+1)}$

$$w_{xy} = \frac{1}{\sum_{q=1}^{d} \left[\frac{\left\| a_x - a_j \right\|}{a_x - d_k} \right]^{\frac{2}{p-1}}} \quad (34)$$

If

$|| W^{(q+1)} - W^{(q)}|| < \varepsilon$ then

STOP

else

return to step 2.

FIS method has proven to be feature correlation logic limits related wto active malware margins by getting actual assigning membership levels to data points capsuled to the next MPNN to classification the RaDSAP; The MLP-NN has incorporated back propagation technique, which determines modification of weights of the neurons during training from the difference between expected and actual outputs. On the IDS, MLP-NN can quickly identify anomalous behavior or intrusion based on the pattern learnt on the traffic flow, therefore helping network security significantly. In equation 35, we compute the hidden layer of the MLP-NN method,

$$I = u_g^b [i_1, i_2, \ldots i_n] \quad (35)$$

Here, I is known as the Input Layer, this equation we represent the input vector of features where each i_x is a feature of the network traffic. In Equation 36, compute the activation function for each neuron in the hidden layer.

$$p_y = \sum_{x=1}^{n} u_{yx} i_x + b_y \text{ and } A_y = f(p_y) \quad (36)$$

Here, p_y is known for the pre-activation function, u is known for the weight, x is known for the input neuron, y is the hidden neuron, and b is the bias term. In equation 2, the p_y is calculated for each neuron in the hidden layer, where $u_{yx} i_x$ is the

weight that connects the x to j. After weights connecting to x to j we compute the activation function 36,In the equation 37the activation function $f(p_y)$ applies a non-linearity to the p_y. Common activation functions include the sigmoid, ReLU, and tanh functions, which are illustrated in equation 37

$$\sigma(p_y) = \frac{1}{1+e^{-p_y}} \quad (37)$$

Here, σ is known for the sigmoid function,

$$f(p_y) = \max(0, p_y) \quad (38)$$

Here, in this equation, we perform the ReLU activation function,

$$f(p_y) = \frac{e^{p_y}-e^{-p_y}}{e^{p_y}+e^{-p_y}} \quad (39)$$

Here, in this equation, we perform the tanh activation function. After perform the activation function in equation 18 we compute the hidden layer. In MLP it has multiple hidden layers,

$$A_y^{(q+1)} = f\left(\sum_{k=1}^{n_q} u_{yk}^{(q)} A_k^{(q)} + b_y^{(q)}\right) \quad (40)$$

Here, q is known for layer, and the output of q becomes the input to the next layer, $q+1$. In this equation, the weighted sum of the outputs from the previous layer is subjected to an activation function. By following the equation, we perform the output layer through equation 41,

$$jk = f\left(\sum_{y=1}^{n_{Q-1}} u_{ky}^{(L)} A_y^{(Q-1)} + b_k^{(Q)}\right) \quad (41)$$

Here, the jk is known for the output, and it is utilized to predict the network. For a binary classification (malicious vs non-malicious), a common activation function is the σ to constrain the output between 0 and 1. After performing the output layer function, we perform the Loss Function D in equation 42,

$$D = -\frac{1}{N}\sum_{x=1}^{N}(j_x \log(\hat{j}_x) + (1-j_x)\log(1-\hat{j}_x)) \quad (42)$$

Here, j_x is known for the actual labels, and $\hat{j}_x$ is known for the predicted outputs. The D quantifies the difference between the j_x and the predicted outputs$\hat{j}_x$. For IDS, binary cross-entropy is commonly used in binary classification tasks. By following equation 43, we perform the backpropagation method, which is used to update the,

$$u_{xy}^{(q)} = u_{xy}^{(q)} - \eta \frac{\partial D}{\partial u_{xy}^{(q)}} \quad (43)$$

Equation 44 adjusts the weights $u_{xy}^{(q)}$ to minimize the D, where η is the learning rate. The partial derivative $\frac{\partial D}{\partial u_{xy}^{(q)}}$ represents the loss rate. In equation 22 we go to classify the traffic as malicious or non-malicious,

$$\hat{j} = \begin{cases} 1, & \text{if} \hat{j}_k \geq 0.5 \\ 0, & \text{if} \hat{j}_k < 0.5 \end{cases} \quad (44)$$

In this equation, the $\hat{j}_k$ id is the threshold value; it is used to classify the traffic as malicious or non-malicious by the binary value of 0,1. The 0 is represent as malicious and the binary value of 1 is the non- malicious. Usually, a threshold of 0.5 is used in binary classification**.**

4. RESULT PERFORMANCE EVALUATION

The proposed AI-powered system has demonstrated significant improvements in performance when benchmarked against existing systems. Key performance indicators such as precision, recall, and true favorable rates consistently surpass those achieved by prior methodologies. This impressive performance stems from the combined strengths of deep learning techniques and refined feature selection processes that address the challenges posed by data shifting and dimensionality. Some parameters used to measure a classification system's effectiveness are classification accuracy, sensitivity, specificity, error rate, and time complexity.

Table 1: Environment and values processed

Simulation Limits	Values
Environment framework	Jupiter notebook
Platform Lang	Python std-Lig

Base repository dataset	RaDSAP
Ach-Logs	98674
Class by ref	Binary class

The comparison algorithms are Support Vector Machine (SVM), wale optimization (WOA), and MLP-GCN, carried out based on the malware detection system. The confusion matrix calculates the following parameters. To work with categorical features in the Darknet dataset, it is necessary to convert non-numeric data to numeric format. This involves performing data transformations to assign a number to each protocol category so that they can be represented numerically.

: IP	Src Port	Dst IP	Dst Port	Protocol	Timestamp	Flow Duration	Total Fwd Packet	Total Bwd packets	...	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Type	packets	target
.11	57158	216.58.220.99	443	tcp	24/07/2021 04:09:48 PM	0	1	1	...	20	20	20	20	20	0	0	Non-Tor	AUDIO-STREAMING	normal
.11	57159	216.58.220.99	443	udp	24/07/2021 04:09:48 PM	0	1	1	...	20	20	20	20	20	0	0	Non-Tor	AUDIO-STREAMING	normal
.11	57160	216.58.220.99	443	tcp	24/07/2021 04:09:48 PM	0	1	1	...	20	20	20	20	20	0	0	Non-Tor	AUDIO-STREAMING	risk
.11	49134	74.125.136.120	443	tcp	24/07/2021 04:09:48 PM	0	1	1	...	20	20	20	20	20	0	0	Non-Tor	AUDIO-STREAMING	risk
.11	34697	173.194.65.127	19305	tcp	24/07/2021 04:09:48 PM	0	1	1	...	20	20	20	20	20	0	0	Non-Tor	AUDIO-STREAMING	risk
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
.11	46461	10.152.152.10	53	tcp	24/02/2018 02:24:58 PM	1	0	0	...	1	1	1	1	1	1	1	Non-Tor	Browsing	normal
.11	53984	54.169.125.186	80	udp	24/02/2018 02:24:58 PM	1	0	0	...	1	1	1	1	1	1	1	Non-Tor	Browsing	normal

Figure 6: Darknet dataset features

In the above figure 6 shows the described dataset feature with a actual rate of margins defined with various standards fields.

Table 2: RAdSAP Parameters

Standard Parameters Taken	Active state parameters
Source IP: Source IP Address **Source Port:** Source Port	Fwd PSH Flags Bwd PSH Flags
Destination IP: Destination IP Address **Destination Port**: Destination Port	Fwd URG Flags Bwd URG Flags
Timestamp: Timestamp for when traffic was sent	Fwd Header Length Bwd Header Length
Protocol: Internet Protocol Version	Fwd Packets/s Bwd Packets/s
FrwdPacketLengthMax FrwdPacketLengthMin FrwdPacketLengthMean FrwdPacket Length Std	Subflow FrwdPackets Subflow FrwdBytes Subflow Bawd Packets Subflow Bawd Bytes
BwdPacketLengthMax BwdPacketLengthMin BwdPacketLengthMean	FWD Init Win Bytes Bwd Init Win Bytes Fwd Act Data Pkts

BwdPacket Length Std	Fwd Seg Size Min
Flow Bytes/s Flow Packets/s	Active Mean Active Std Active Max/ Active Min
Protocol: Internet Protocol Version	SYN Flag Total RST Flag Total PSH Flag Total ACK Flag Total URG Flag Total CWE Flag Total ECE Flag Total Down/Up Ratio

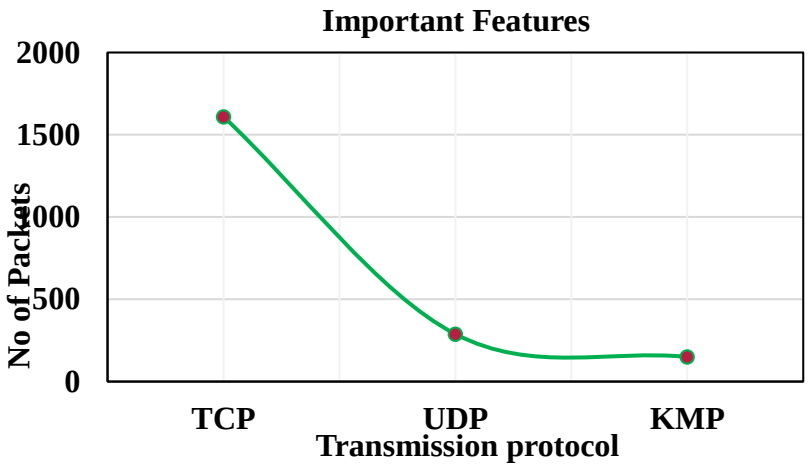

Figure 7: Screenshot feature-based ICSSIT-HCDNN optimization

In the above figure 7, the features selection using swarm intelligence carried out by the standard network protocols affected by different variation leads from timestamp protocols. Though out packet streaming in hist level protocol variation and ints length are progressed.

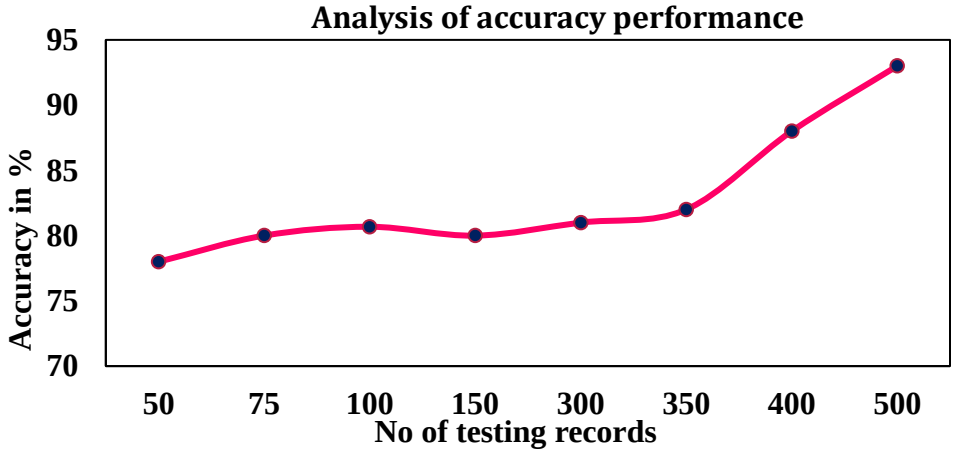

Figure 8: Analysis of Accuracy

Figure 8 described as proposed system accuracy in different level of true positive, and false neative omabination inder the lvel described by acurayc rate in proposed system 0.93%, comparing the other methods are MLP-GCN is 0.88%, WOA is 0.80% and SVM is 0.77%. Proposed method shows the better accuracy compared to the previous methods.

Table 3: Enactment on intrusion detection accuracy vs. num of services

IDS Accuracy in % vs Number of Services			
Comparison techniques/ services	50 iteration	100 interaction	150 iteration
SVM	69.8	72.9	77.2
WOA	76.2	78.6	80.7
MLP-GCN	78.7	82.4	86.9
ICSSIT-HCDNN	90.7	95.8	95.5

Table 3 displays the accuracy of IDS, considering the number of services used and the techniques employed, including SVM, WOA, MLP-GCN, and the proposed Ids-based feature analysis model AFIS.

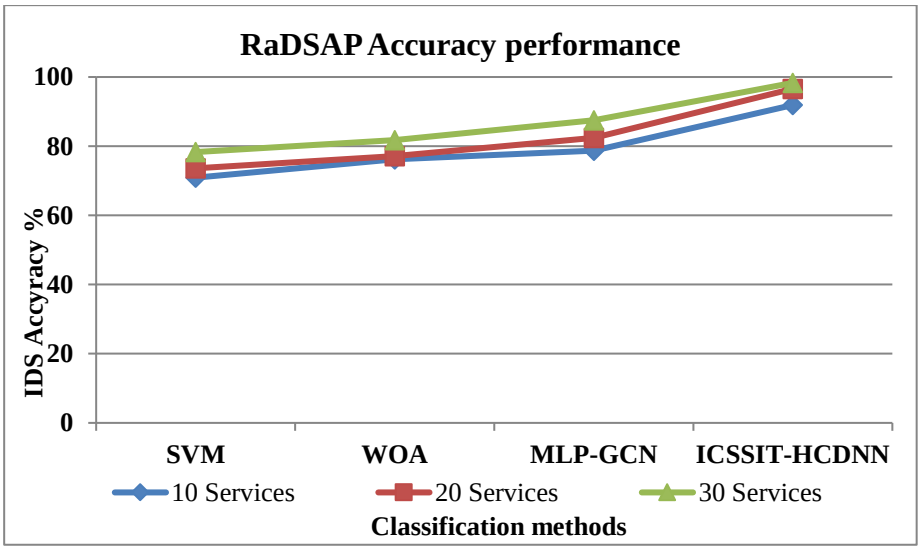

Figure 9: Impact of IDS accuracy performance

In Figure 9, it can be demonstrated how the accuracy of intrusion detection classification is affected by different services (10, 20 and 30). The ICSSIT-HCDNN method proposed in this study achieves a remarkable 98.3% accuracy for 30 services. This is significantly better than the previous SVM, which completed only 78.3%, and the WOA and MLP-GCN techniques, with 81.8% and 87.8%, respectively. Overall, the proposed method outperforms other designs in terms of accuracy.

Table 4: Impact of Sensitivity performance

Comparison methods/ services	10 Services	20 Services	30 Services
SVM (%)	69.6	72.5	77.6
WOA (%)	73.5	76.3	80.1
MLP-GCN (%)	77.6	80.9	86.3
ICSSIT-HCDNN (%)	88.4	93.1	94.9

In Table 4, the performance impact of the proposed sensor is outlined in comparison to previous techniques.

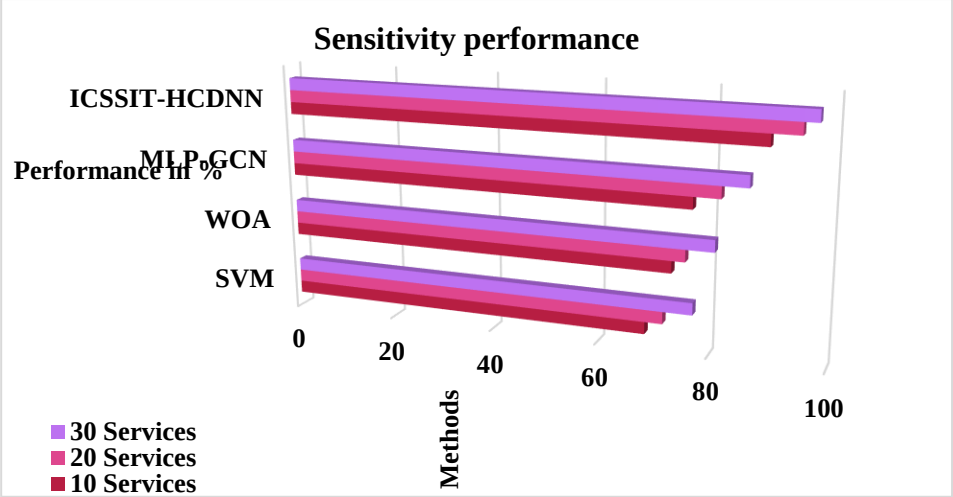

Figure 10: Analysis of sensitivity

Figure 10 displays the sensitivity performance of IDS detection utilizing the ICSSIT-HCDNN- algorithm. The proposed algorithm achieved a sensitivity performance of 96% for 30 services. On the other hand, other algorithms such as SVM, WOA, and MLP-GCN only managed to reach 76%, 81%, and 85% sensitivity performance, respectively, for the same number of services.

Table 5: Impact of Specificity Performance

Comparison methods/ services	10 Services	20 Services	30 Services
SVM (%)	72.3	74.6	78.3
WOA (%)	78.6	76.9	83.5
MLP-GCN (%)	79.3	82.6	86.9
ICSSIT-HCDNN (%)	90.6	95.1	96.7

Table 5 analyses performance metrics for 10, 20, and 30 services. The proposed technique produces superior outcomes compared to previous methods.

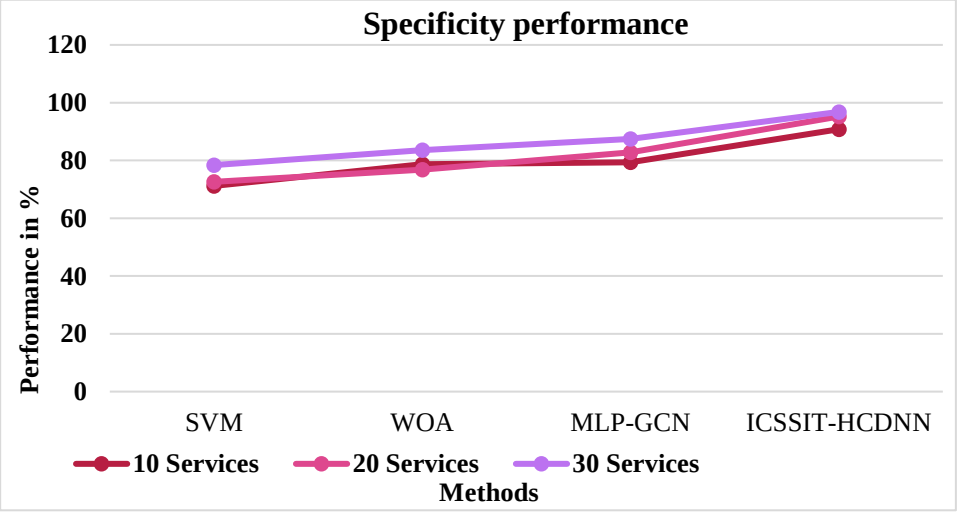

Figure 11: Analysis of Specificity Performance

Figure 11 displays the performance analysis of the offered technique and provides a comparison to previous methods. The ICSSIT-HCDNN algorithm proposed has a specific performance of 97% for 30 services. Meanwhile, the SVM algorithm for typical efficiency for 30 services results in a 77% specific performance, the WOA algorithm in an 82% specific version, and

the MLP-GCN algorithm in an 86% specific performance.

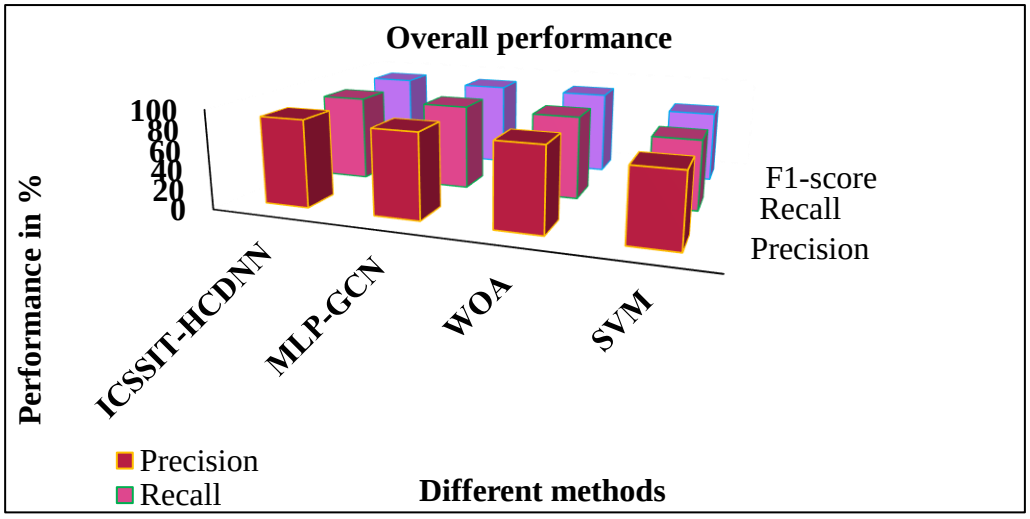

Figure 12: Analysis of overall performance accuracy

Figure 12 described based on the performance matrix calculated based on the confusion matrix (TP, FP, and FN) estimations depend on the dataset truth value rate. In the ICSSIT-HCDNN analysis of the proposed method, the precision score is 0.94%, the recall score is 0.89%, and the F1 score ID is 0.92%. FLSP provides 0.90% precision, 0.88% recall, and 0.89% f1 score compared to previous techniques. In addition, it has 0.88% precision, 0.86% recall and 0.84% f1 score with 0.81% MDDLF-IoT precision rate and DR7 to 0. The F1 score is 0.80, the BR-IoT accuracy score is 0.rec79%, the rate is 0.74%, and the f1 score is 0.71%.

Table 6: Analysis of false classification ratio

False Classification Ratio in % vs. No of Services			
Comparison methods/ services	**10 Services**	**20 Services**	**30 Services**
SVM	27.6	26.3	19.5
WOA	24.4	22.5	15.8
MLP-GCN	19.3	17.7	11.5
ICSSIT-HCDNN	5.9	2.6	1.4

Table 6 analyzes the misclassification rate for the suggested technique compared to the performance of previous methods.

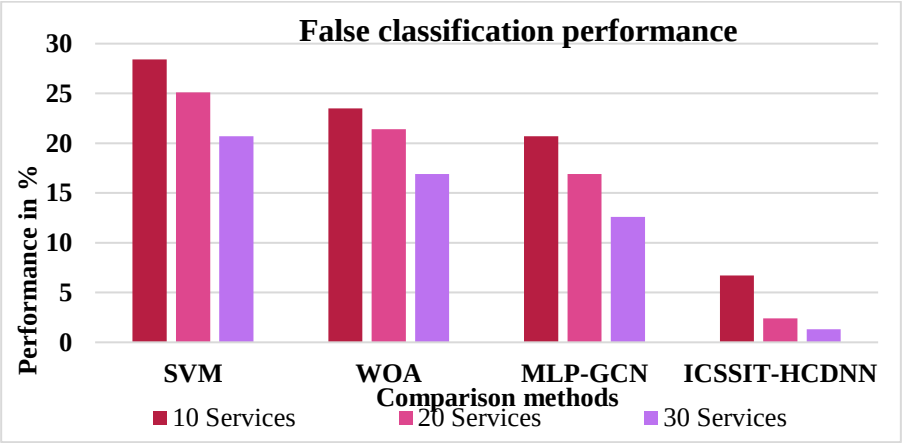

Figure 13: Impact of false classification ratio

Figure 13 shows the impact of the misclassification rate on IDS performance for services at 10, 20, and 30. X-axis represents the corresponding model, and Y-axis shows its slope. The performance of each technique is remarkable; SVM has a misclassification rate of 20.7%, while our proposed ICSSIT-HCDNN model has a misclassification rate of 1.3% for 30 services. The WOA method has a rate of 16.9%, and the MLP-GCN method has a rate of 12.6%.

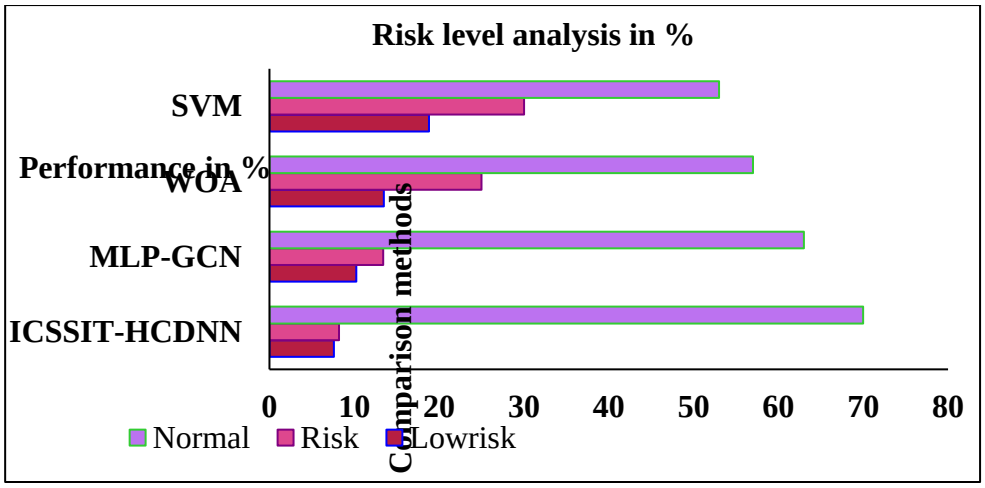

Figure 14: Analysis of Multi Scale Features Risk Level

Figure 14 described as feature risk progressed in ransomware properties with different level of approach carried in comparison and the proposed system produce high performance.

Table 7: Impact of time complexity performance

Time complexity in seconds vs No of Services			
Comparison methods/ services	**10 Services**	**20 Services**	**30 Services**
SVM	61.5	67.3	68.1
WOA	49.7	51.6	57.4
MLP-GCN	30.9	36.4	41.2
ICSSIT-HCDNN -	17.4	21.4	28.2

Table 7 outlines the performance of various services and their time complexity. Specifically, for ITS classification, SVM takes 67.2 seconds, WOA takes 58.6 seconds, and MLP-GCN takes 41.2 seconds. However, the proposed AFIS only takes 28.3 seconds for IDS classification.

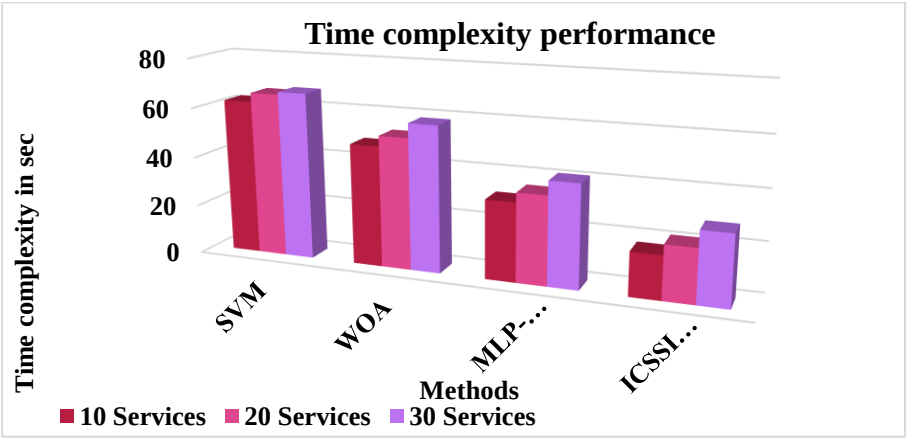

Figure 15: Result of time complexity performance

Figure 15 shows the time complexity performance results of a proposed service-specific payload inference analysis model using the ICSSIT-HCDNN technique. This model is compared to SVM, WOA, and MLP-GCN methods. The x-axis shows the compared methods, and the y-axis shows their time-complexity performance in seconds. Remarkably, the proposed method provides faster results than the previous techniques.

Table 8. Performance on various measures

Comparison methods/ services	Detection Rate %	False Ratio %	Time Complexity in sec
SVM	79.8	19.1	61.6
WOA	82.1	17.3	47.2
MLP-GCN	87.8	11.4	30.4
ICSSIT-HCDNN	97.2	1.2	17.8

The performance of the suggested AFIS can be observed in Table 8, which displays various metrics such as detection rate, error rate, and time complexity. The proposed AFIS technique has superior performance compared to commonly used methods. Table 7 described the Response ratio and time complexity based on the source to-destination data transfer without traffic.

$$Response\ ratio(time) = \frac{Data\ size}{bit\ rate} \quad (44)$$

$$Total\ Time\ complexity = [O\{\propto \{O(N) + O(N) + O(t) + O(t) + O(x) + O(xy) + O(x^2y)\}\}]$$

$$= O(\propto (N + T + x^2y \quad (45)$$

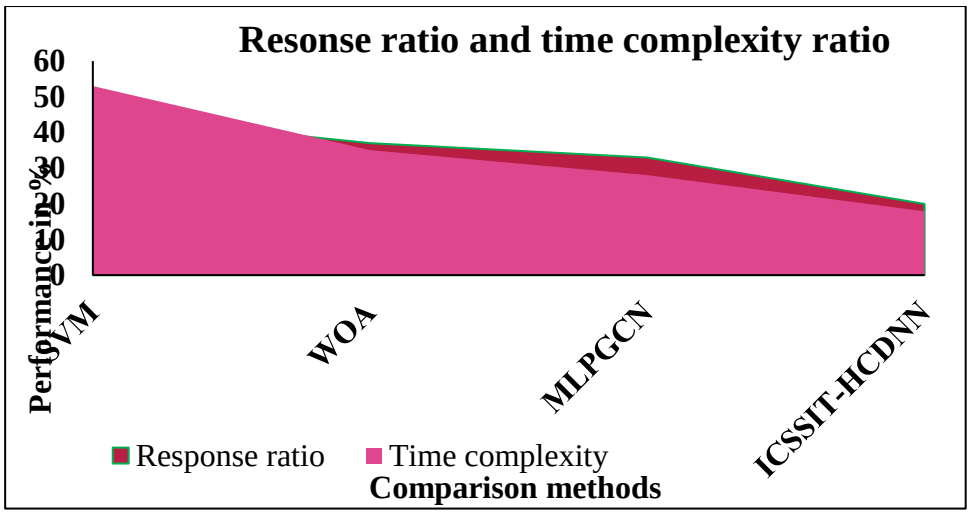

Figure 16: Analysis of Response Ratio

Figure 16 described the Response ratio produced from different methods to evaluate the identification rate , ICSSIT-HCDNN is 36s immediate response for Central Processing Unit (CPU), and compared to previous methods takes a time MLPGCN is 45s, DRL is 49s, WOA is 52s and SVM is 55s. Time complexity is ICSSIT-HCDNN is the 20s, MLP-GCN is 32s, WOA is 38s, SVM is 45s.

5. CONCLUSION

In conclusion, the increasing interconnectivity brought about by the IoT in the healthcare sector underscores the urgency of robust cybersecurity mechanisms, particularly in the face of rising ransomware threats. Traditional methodologies for ransomware detection are inadequate, given the complexity of modern cyber-attacks. By leveraging advanced techniques such as ICSIT and HCDNN, our proposed system offers a pioneering solution that enhances the detection and analysis of ransomware behaviors, ultimately boosting the security of sensitive healthcare data. The integration of AI into cybersecurity frameworks not only promotes efficiency but also ensures that healthcare institutions can deliver care without the debilitating constraints imposed by cyber vulnerabilities. As healthcare continues to evolve with technology, ICSSIT-HCDNN embracing innovative solutions for cybersecurity will be essential in safeguarding patient data and maintaining the integrity of healthcare services.

REFERENCES

[1] Sheen, S., Asmitha, K. A., & Venkatesan, S. (2022). R-Sentry: Deception based ransomware detection using file access patterns. Computers and Electrical Engineering, 103, 108346. https://doi.org/10.1016/j.compeleceng.2022.108346

[2] McIntosh, T.; Kayes, A.; Chen, Y.P.P.; Ng, A.; Watters, P. Ransomware mitigation in the modern era: A comprehensive review, research challenges, and future directions. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–36.

[3] Zhang, Chongzhen, et al. "A novel framework design of network intrusion detection based on machine learning techniques." Security and Communication Networks 2021.1 (2021): 6610675.

[4] Shankar, D., et al. "Deep analysis of risks and recent trends towards network intrusion detection system." *International Journal of Advanced Computer Science and Applications* 14.1 (2023).

[5] Sladkova, Polina, et al. "Adaptive deep learning-based framework for ransomware detection through progressive feature isolation." (2024).

[6] Alraizza, A.; Algarni, A. Ransomware Detection Using Machine Learning: A Survey. Big Data Cogn. Comput. 2023, 7, 143. https://doi.org/10.3390/bdcc7030143.

[7] Ali, Rahman, et al. "Deep learning methods for malware and intrusion detection: A systematic literature review." Security and Communication Networks 2022.1 (2022): 2959222.

[8] Yamany, B., Elsayed, M. S., Jurcut, A. D., Abdelbaki, N., & Azer, M. A. (2021). A New Scheme for Ransomware Classification and Clustering Using Static Features. Electronics, 11(20), 3307. https://doi.org/10.3390/electronics11203307.

[9] Khammas, B. M. (2020). Ransomware Detection using Random Forest Technique. ICT Express, 6(4), 325-331. https://doi.org/10.1016/j.icte.2020.11.001.

[10] Zahoora, U., Khan, A., Rajarajan, M., Khan, S. H., Asam, M., & Jamal, T. (2022). Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier. Scientific Reports, 12(1), 1-15. https://doi.org/10.1038/s41598-022-19443-7.

[11] Hwang, J., Kim, J., Lee, S. et al. Two-Stage Ransomware Detection Using Dynamic Analysis and Machine Learning Techniques. Wireless Pers Commun 112, 2597–2609 (2020). https://doi.org/10.1007/s11277-020-07166-9.

[12] Lee, J., & Lee, K. (2022). A Method for Neutralizing Entropy Measurement-Based Ransomware Detection Technologies Using Encoding Algorithms. Entropy, 24(2), 239. https://doi.org/10.3390/e24020239.

[13] Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2021). Differential area analysis for ransomware attack detection within mixed file datasets. Computers & Security, 108, 102377. https://doi.org/10.1016/j.cose.2021.102377

[14] Brinkley, Yenisel, Daniel Thompson, and Nicholas Simmons. "Machine learning-based intrusion detection for zero-day ransomware in unseen data." (2024).

[15] Zhang, Xueqin, Jiyuan Wang, and Shinan Zhu. "Dual generative adversarial networks based unknown encryption ransomware attack detection." IEEE Access 10 (2021): 900-913.

[16] Y. Yang, K. Zheng, B. Wu, Y. Yang and X. Wang, "Network Intrusion Detection Based on Supervised Adversarial Variational Auto-Encoder with Regularization," in IEEE Access, vol. 8, pp. 42169-42184, 2020, doi: 10.1109/ACCESS.2020.2977007.

[17] Sai Chaitanya Kumar, G., Kiran Kumar, R., Parish Venkata Kumar, K., Raghavendra Sai, N., & Brahmaiah, M. (2024). Deep residual convolutional neural Network: An efficient technique for intrusion detection system. Expert Systems with Applications, 238, 121912. https://doi.org/10.1016/j.eswa.2023.121912.

[18] Arif, J. M., Ab Razak, M. F., Awang, S., Tuan Mat, S. R., Nadiah Ismail, N. S., & Firdaus, A. (2021). A static analysis approach for Android permission-based malware detection systems. PLOS ONE, 16(9), e0257968. https://doi.org/10.1371/journal.pone.0257968

[19] Revathy, G., P. Sathish Kumar, and Velayutham Rajendran. "Development of IDS using mining and machine learning techniques to estimate DoS malware." International Journal of Computational Science and Engineering 24.3 (2021): 259-275.

[20] Rajest, S S. "Application of the Catboost Classifier for the Detection of Android Ransomware," Central Asian Journal of Mathematical Theory and Computer Sciences 2024, 5(5), 476-486.

[21] Urooj, Umara, et al. "Addressing Behavioral Drift in Ransomware Early Detection Through Weighted Generative Adversarial Networks." IEEE Access (2023).

[22] Zanoramy, Wira, et al. "Ransomware early detection using machine learning approach and pre-encryption boundary identification." Journal of Advanced Research in Applied Sciences and Engineering Technology 47.2 (2024): 121-137.

[23] Hammood, Dalal Abdulmohsin. "A hybrid system based on machine learning and PSO for network intrusion detection." AIP Conference Proceedings. Vol. 3232. No. 1. AIP Publishing, 2024.

[24] Sladkova, Polina, et al. "Adaptive deep learning-based framework for ransomware detection through progressive feature isolation." (2024).

[25] Kaushik, Baijnath, et al. "Performance evaluation of learning models for intrusion detection system using feature selection." Journal of Computer Virology and Hacking Techniques 19.4 (2023): 529-548.