FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 18, September, 2026, pp 10 – 15
DOI: https://doi.org/10.33003/fjs-2026-1018-5854

Comparative Evaluation of Selected Machine Learning Classifier Algorithms for Ransomware Detection in Computer Networks

***Musa Abdulhakeem Yusuf, Kamil Kayode Saka and Ismail Jamiu Okunlola**

Department of Computer Science, Faculty of Computing, Engineering and Technology, Al-Hikmah University, Ilorin, Kwara State, Nigeria.

*Corresponding authors' email: a08079835696@gmail.com

Received: 31 Aug. 2026
Accepted: 01 Sep. 2026
Published: 16 Sep. 2026

ABSTRACT
Network based extortion is the first order risk that has become the computer network, and has become a vital part of how organisations communicate, record and provide services. Ransomware wreaks havoc on services, encrypts files, finances, reputational and regulatory damage and signature-based ransomware defenses don't work very well for faster detection. In this study, the objective was to determine the best detector from four popular supervised classifiers for the case of class imbalance and feature redundancy. A quantitative experimental design was used. The UNSW-NB15 benchmark contained approximately 257,000 labelled records and 49 features was obtained. The records were cleaned, encoded and scaled, the class imbalance was corrected using an extreme gradient boosting model with a weighted loss function; the dimensionality was reduced using principal component analysis. The reduced set was then split in the ratio of 70:30 and then classified by four classifiers (SVM, RF, ANN and DT). The reproducibility was calculated with a fixed partitioning and an invariant pipeline and a peer reviewed benchmark. The accuracy rate of the neural network was 99.50 per cent and F1 score of 99.52 per cent, better than the other classifiers, random forest (98.38), support vector machine (98.04) and decision tree (96.00). The study is based on a like for like comparison, the imbalance correction and feature extraction are preserved and only the type of classifiers are changed. The choice of classifiers will have effect on the performance of the detection, however, the difference between good models-preprocessed models is not significant.

Keywords: Ransomware Attack, Class Imbalance, Machine Learning Classifiers, Network Security, Principal Component Analysis, Ransomware Detection

INTRODUCTION

Networked computing is widely used in companies, hospitals, universities and public agencies on a daily basis. Previously documents were in a filing cabinet, now on a server that's accessible from anywhere, people hand documents from one office to the next, and deals are struck between banks. This efficiency is only afforded by the same connectivity that can put an entire organisation at risk if it's a vulnerable endpoint. The biggest threat to impact the use of this type of exposure, monetarily, has been ransomware. It can either encrypt data or deny access to systems and require a ransom to regain access, thus becoming an immediate concern of operational continuity (Oz et al., 2022).

The loss is more than just the ransom. The damages of lost production, unmet clinical services and loss of reputation due to the release of information are often more than monetary (Al-rimy et al., 2018). There are now more targeted attacks. Early campaigns were broader and more indiscriminate, whereas modern campaigns target specific documents and/or groups of documents, as well as disable backups before starting to encrypt data, and also threaten to publish exfiltrated records (Oz et al., 2022). Legacy equipment, few devices and long patching times are some of the reasons that industrial and embedded environments are considered some of the most vulnerable (Al-Hawawreh et al., 2024; Serror et al., 2021).

It's hard, if not impossible, for the traditional countermeasures to catch up with this trend. A signature based intrusion detection system (IDS) checks for known signatures present in the system will be silent on the absence of signatures known in their catalogue (Alraizza & Algarni, 2023). The controls are never followed for polymorphic families which alter their own code while infecting, and those zero-day variants that haven't got any signature at present (Urooj et al., 2022).

Machine learning is based on a different principle. While a trained classifier doesn't necessarily need to match known artefacts, it needs to learn the statistical regularities that differentiate malicious and benign activity and then generalise them to samples that it has not yet encountered before (Ispahany et al., 2024). Examples of techniques that have been used with good success in malware and intrusion detection are: support vector machines (SVM) (Albin Ahmed et al., 2023), random forest (RF) (Khan et al., 2020), artificial neural networks (ANN) and decision trees (DT) (Albin Ahmed et al., 2023; Khan et al., 2020).

There are two promises that need to be fulfilled with regard to this obstacle. This fact is related to the highly unbalanced distribution of detection corpora, where in every real capture, there would be far more numbers of benign activities than there would be malicious activities, so that the learner could have a high accuracy rate, but will likely not produce the minority class (malicious) that is relevant (Zahoora et al., 2022). Moreover, many redundant or weakly informative features (Jolliffe & Cadima, 2016) exist in network telemetry that could be a part of the cost of training and the possible overfitting. Research papers that report a false positive for a "headline", but don't address both problems or test the algorithm on a variety of corpora and with various preprocessing, leave the practitioner without a good reason to

select the algorithm. Some research papers produce incorrect results for a headline, and do not take into account both issues (or compare with other algorithms trained on other corpora, with other preprocessing). This means that the researcher is unable to choose a specific algorithm.

This paper makes a contribution to counter that. The aim was to develop a more effective detection system using the machine learning classifiers chosen and to identify the most effective of these classifiers for the constant pre-processing system. Data was preprocessed by using extreme gradient boosting technique to overcome the problem of class imbalance; principal component analysis was used for feature extraction; support vector machine, random forest, artificial neural network and decision tree were the classifiers used to evasluate and compare the models; and the accuracy, precision, recall, F1-score and computational efficiency were the evaluation parameters.

Review of Related Works

Ransomware and Signature-Based Defense

Ransomware operates on a distinct economic model from other malware families. It is not used for stealth or for the purpose of data exfiltrating for intelligence uses; rather it is intended to create an immediate cost to the victim either by encrypting the victim's data and/or by denying access to the data, and then demanding a ransom (Al-rimy et al., 2018). The process of typical attack involves delivery, deployment, key exchange, encryption and extortion, and the first stages of the attack are the only ones where recovery is, more or less, feasible – after encryption, recovery is all but impossible, without the decryption key.

The Signature Based IDS remains the main method of protection used by most organisations, and is very reactive. They are able to find known families of well known-bytes and behavioural signatures and consequently stay quiet to the polymorphic variants that modify their code and to the zero day families that haven't ever been catalogued (Alraizza & Algarni, 2023; Urooj et al., 2022). Static analysis of binaries is simple to implement, but can be easily defeated by using packing and obfuscation and cannot show what occurs during runtime; dynamic analysis of binaries can provide more insights into what happens at runtime but the execution environment must be controlled and can't be easily scaled to all endpoints. However, network based detection wouldn't need any agents on the individual hosts and would be more suitable for being deployed in enterprises (Vehabovic et al., 2022). The aim of this study is thus flow level classification of network telemetry.

Machine Learning Approaches and Their Limitations

The current paradigm, ransomware and intrusion detection, is the supervised machine learning model (SML) that can generalise the statistical regularities in addition to exact match signatures (Ispahany et al., 2024). Four illustrative studies give insights on progress and methodological constraints of the practical implementation.

Khan et al. (2020) proposed a novel digital DNA sequencing engine, called DNAact-Ran, which re-shaped the selected operational-code feature to that of the k-mer frequencies vectors and classified the sequences using active learning. It is tested on a small number of 582 ransomware samples and 942 goodware samples, and got high precision, recall and accuracy. However, the study did not take into account the problem of imbalance and the feature transformation method was proprietary making it hard to replicate. Furthermore, there were no controlled comparisons to standard classifiers, with the same preprocessing.

Albin Ahmed et al. (2023) have developed a model using Decision Tree, Support Vector Machine (SVM), KNN, Ensemble and NN algorithms, 392 035 traffic data from Android and 10 ransomware families. The decision tree was obtained with 97.24 % accuracy and support vector machine was obtained with 100 % recall with the complete set of features. Unfortunately, no experiments with explicit imbalance correction, dimensionality reduction (apart from selecting the features) were done, nor was there any attempt to isolate the effect of the classifier from the rest of the pipeline.

Zahoora et al. (2022) proposed a cost-sensitive Pareto ensemble which focuses on learning cost balance between false positive and false negative and a contractive autoencoder that learns some compressed representations to tackle the zero-day ransomware problem. The framework showed a better detection of ‘blind' variants. But it is complicated in some ways: It relies on deep unsupervised feature extraction and a special ensemble, as well as the evaluation simultaneously tests several design decisions – which makes it difficult to pinpoint performance improvements from one change.

Irankunda et al. (2025) found that Generative Adversarial Networks, Autoencoders, Long-Short Term-Memory Networks and Convolutional Networks when applied to the portable-executable header features and IoT-23 corpus, are effective in the detection of ransomware in the IoT space. The model with maximum accuracy was LSTM and CNN model which was 96.98 %. The current model is a sequence model, different from the previous model, it adopts the features of the executable level of the model, rather than the features of the network-flow level, and does not use the weighted loss function method to solve the problem of unequal class.

Besides, light weight ensemble methods are also recently explored for the classification of ransomware. Recently, Akuboh et al. (2026) achieve an accuracy of 99.10% using a combination of ensemble of soft voting method and recursive feature elimination and principal component analysis on the ransomware subset of CIC-AndMal2017.

All of these works have proven effective in achieving a high accuracy rate of malicious or benign behaviour identification with machine learning approach. While they are simultaneously, they have three common drawbacks. Firstly, class imbalance is not addressed or partially addressed and the accuracy reported might be overestimating the accuracy of the minority class (malicious). Secondly, in most networks, high dimensional (or redundant) features are not typically reduced, which may result in loss of information and increase in computation. Thirdly, the experimental conditions (corpus, preprocessing, partitioning and evaluation metrics) are also changed simultaneously, making it hard to distinguish the difference in performance due to the type of classifiers and the difference in other experimental conditions (Ispahany et al., 2024; Alraizza & Algarni, 2023).

The Gap Addressed by the Present Study

The only experimental variable studied in the present study is the classifiers. The same random split (70:30) of the flow corpus UNSW-NB15 is used to train all the models, the same evaluation metrics (accuracy, precision, recall, F1-score and area under the ROC curve) are reported, the same weighted loss function is used to handle the class imbalance issue, and the same redundant features are removed using principal component analysis (PCA). Four popular supervised classifiers, namely support vector machine, random forest, artificial neural network and decision tree are compared under these controlled conditions. Now we have the right processing

pipeline and the only way that a difference in detector performance could be achieved would be by the inductive bias of the detector (classifier) and not because of other design considerations. The controlled comparison is a direct outgrowth of the methodological mismatch in the most recent surveys, and offers a definite decision-point for practitioners to choose a detector.

MATERIALS AND METHODS

Model Design

A quantitative experimental design was used in this study. The design is appropriate since the research question is asking about which algorithm works best in the specific condition of the experiment in which the condition of training is balanced. All such models were thus created, trained and tested in a controlled pipeline, which was created in python using the Pandas and Scikit-learn libraries. The framework is shown in the figure 1.

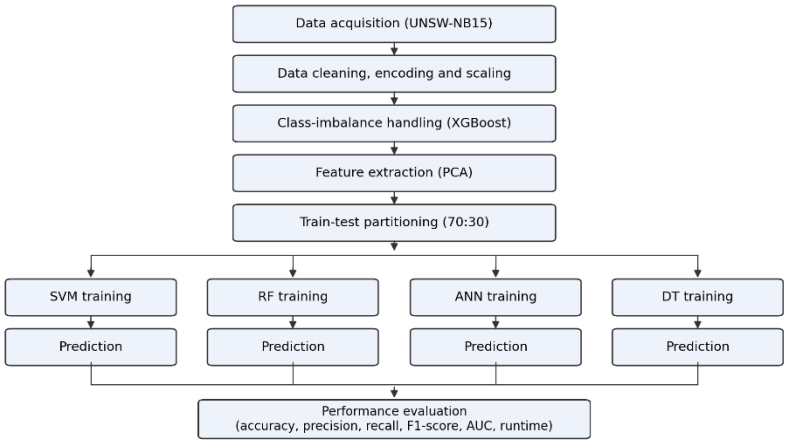

Figure 1: Methodological framework of the detection pipeline

Data Acquisition

The UNSW-NB15 corpus was downloaded in the comma separated format from a public repository of data and was used. It has been developed in IXIA PerfectStorm traffic generator with the background traffic and attack traffic generated in the 21st century not the KDD99 attack traffic of Moustafa & Slay (2015, 2016). The size of the working set was around 257,000 network-flow records, with each record having 49 features that were labelled as either benign or malicious. There are attack types (including DDoS, exploits, reconnaissances and generic attacks) and a detector can be measured against an attack family, but not just against a single attack type. The features can be classified into four groups as summarised in Table 1.

Table 1: Feature Categories in the UNSW-NB15 Corpus

Feature category	Description
Basic features	Packet-level attributes such as protocol type, service and connection duration
Content features	Attributes describing packet content and payload characteristics
Time-based features	Traffic statistics computed over temporal windows
Flow-based features	Statistical properties derived from complete network flows

The corpus was selected for three reasons: it's peer-reviewed and has been widely used to allow for external comparison; it's labelled at the flow level, which is consistent with a network-based detection approach; the feature groups are packet-based and temporal, as the propagation and command-and-control activity of ransomware is observed in traffic.

Data Preprocessing and Imbalance Correction

Cleaning preceded modelling. Values for missing numbers were filled in with the mean or median of values for the attribute, based on the distribution of the attribute, while categorical values were filled in with the mode of the attribute, and attributes with too many missing numbers were not used. Duplicate records have been eliminated as repeated flows make it look like performance is higher than it actually is. Outliers were identified using the interquartile range rule and checked before any cap/cut was applied, because they might be an extreme value of the signal (the security telemetry data) and not noise. Values that could be identified as collection artefacts were capped/removed. The categorical attributes (those with an ordering) were labelled encoded and the attributes without an ordering were one-hot encoded, preventing an arbitrary ordering on the nominal attributes being assumed. The features were standardised and this is important for the support vector machine and the neural network as they are sensitive to the magnitude of the difference in the features.

Then, the extreme gradient boosting technique was applied in order to solve the class imbalance problem. The scale-positive-weight parameter is a ratio of the negative to positive examples, and increases the penalty for misclassifying the minority class, meaning malicious flows will have an impact on the objective function based on how important they are, not how often they occur (Chen & Guestrin, 2016). The model is additive and in the subsequent rounds, the minority instances become more and more important because every tree is fitted to the previous round's residual error. This process can be constrained by the use of first and second order regularisation terms, which help to prevent overfitting. The method was preferred to a synthetic oversampling method (synthetic minority data) or a minority sampling method (actual minority data), as this latter approach would require the collection of minority data from the source, which was not feasible.

Feature Extraction

Principal component analysis was used to analyse the balanced and standardised data. To reduce the data to a lower dimensional space, the covariance matrix was calculated, the eigenvalue decomposition was performed, then the eigenvectors were sorted by the largest eigenvalues, and the eigenvectors with the largest eigenvalues were kept and used to transform the data (Jolliffe & Cadima, 2016). The correlation between the related flow statistics will be eliminated by the transformation, thus avoiding counting the same type of statistics twice; the variance that contains the discriminative signal will remain.

Classification and Evaluation

The reduced data has been randomly divided into a 70 per cent training data set, and a 30 per cent testing data set. The same partition was used for training four classifiers. For high dimensional spaces, the separating hyperplane identified by the SVM is the one with the largest margin between the classes. Random forest is based on the idea of generating decision trees based on random samples and assuming that the trees are the majority vote process, with variance of each individual tree being minimized (Breiman, 2001). Using artificial neural network, the non-linear separating relationship is used, and the non-linear activation function and weighted layer are used to map from input to output; the weight is adjusted by backpropagation. The decision tree is a recursive partition of the feature space to minimize Gini impurity and is used as the baseline to be interpreted.

The performance was evaluated using the accuracy, precision, recall, F1-score and area under receiver operating characteristic curve, all from the confusion matrix, and the processing time measured by running the program to extract the features. Recall is a key consideration in this application: a false positive can not only disrupt legitimate work, but a false negative allows an encryption event to be passed, which is not equally costly with a false positive.

RESULTS AND DISCUSSION

Result of Feature Extraction

Table 2 reports the time required by the dimensionality-reduction stage.

Table 2: Computational Cost of Feature Extraction Using Principal Component Analysis

Stage	Time taken (seconds)
Training (fitting the transformation)	1.230
Testing (transforming unseen data)	1.425

The transformation required 1.230 seconds, and unseen data needed 1.425 seconds for the transformation. The difference is 0.195s, which is due to the extra effort that goes into projecting new observations onto the retained components. In absolute numbers, both are small, confirming that the extraction stage does not introduce any meaningful latency penalty, and thus would not on its own pose a deployment barrier in a scenario where flows need to be scored as they are received.

Comparative Classifier Performance

Table 3 sets out the performance of the four classifiers on the held-out test partition.

Table 3: Comparative Performance of the Machine Learning Classifiers

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC
Artificial neural network	99.50	99.19	99.85	99.52	1.000
Random forest	98.38	97.57	99.34	98.45	0.999
Support vector machine	98.04	97.26	99.02	98.13	0.997
Decision tree	96.00	96.32	95.94	96.13	0.960

The artificial neural network achieved the best result across all measures with 99.50 per cent accuracy, 99.19 per cent precision, 99.85 per cent recall and an F1-score of 99.52 per cent. Among these figures, its recall is the most important as it highlights that very few malicious flows were not classified. The random forest model achieved 98.38 per cent accuracy, the support vector machine at 98.04 per cent and the decision tree at 96.00 per cent. In the accuracy comparison, shown in figure 2, the data sets are presented in a different format.

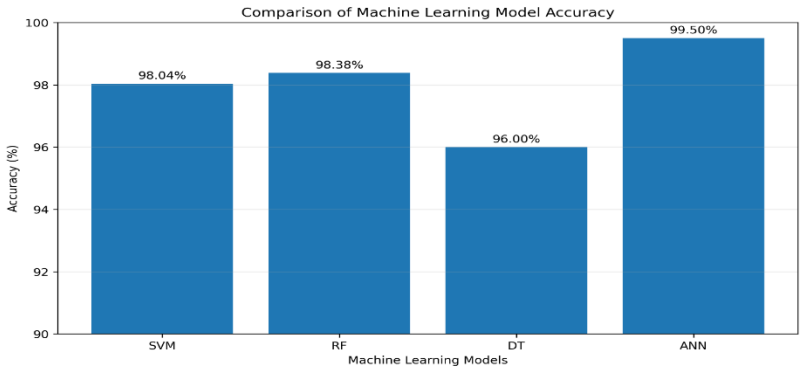

Figure 2: Comparative accuracy of the four classifiers

It is not merely a function of one particular measure, and is consistent across measures. They are also close together: the gap between the best and worst models is only 3.5 percentage points, and all four models are above 95 per cent. This, in

itself, is a finding: even the simplest classifiers are elevated to a level that would be useful in practice by the preprocessing pipeline.

Comparison with Existing Studies

Table 4 situates the results against two recent studies that report accuracy on comparable tasks.

Table 4: Comparison of the Present Results with Selected Recent Studies

Study	Approach and data	Best reported accuracy (%)
Albin Ahmed et al. (2023)	Decision tree, support vector machine, nearest neighbour and neural models applied to 392,035 Android traffic records	97.24
Irankunda et al. (2025)	Generative adversarial networks, autoencoders, long short-term memory and convolutional networks applied to executable header features and the IoT-23 corpus	96.98
Present study	Extreme gradient boosting for imbalance and principal component analysis for feature extraction, with support vector machine, random forest, neural network and decision tree classification of UNSW-NB15 flows	99.50

The models built here achieve better rates of accuracy than do both of the comparators. It must be said that the three studies are based on different corpora and different feature representations, so the figures do not necessarily reflect the superior performance of the present pipeline but rather a level of competition.

Discussion

The following observations are made from these results. The first one relates to the order of the classifiers. While the sequence of univariate splits of the space used by the single decision tree is very efficient, the neural network and the random forest (which model interactions among features using either hidden layers or ensemble aggregation) proved more effective. The margin is relatively small but steady and it indicates that discriminative information in network telemetry is not just contained within individual attributes, but is also embedded in complex combinations of attributes. It is consistent with this reading that the linear-margin method, the support vector machine, lies in between the two extremes.

The second is about the impact of the preprocessing. The four classifiers had all achieved accuracies above 95 per cent with the worst being close to the standard reported in the literature for tuned classifiers. The compression of the range of performances is likely to be due to these two stages, since they remained constant across the different conditions. The takeaway is that while the former may be more profitable, it might also be more likely to be useful than the latter, which is what Ispahany et al. (2024) argue for methodology rigor rather than novelty.

The third is about the deployment of the organisation. The ability of the security team to detect is not the only criterion. The decision tree is the least accurate but yields rules which an analyst can review and explain to management and thus has value when a detection needs to be explained prior to isolating a system. The neural network has the advantage of having better discrimination ability and less transparency. If regulatory accountability is in question, it is a governance issue, and not just a technical one, to determine the trade-off between accuracy and explicability.

These results are tempered by four limitations. The UNSW-NB15 is a general intrusion benchmark and not a ransomware-specific one; therefore, the models are able to distinguish between malicious and benign network activities, and ransomware detection follows from that rather than being shown against labeled ransomware families – the most important limitation on the conclusions. Traffic is artificially generated in a controlled laboratory setting and it cannot be representative of the variability of production networks. The estimates were not based on cross validation, but on a single holdout partition, so they do not have a confidence interval. Lastly, very high scores require careful interpretation: scores at the top of a benchmark corpus often do not perform as well on real traffic, and therefore shouldn't be viewed as a good operational benchmark.

CONCLUSION

In this work, four supervised classifiers have been compared for the detection of malicious network activity in a pipeline where the class imbalances were addressed by extreme gradient boosting and dimensionality reduction by principal component analysis. The random forest, the support vector machine and the decision tree performed better than the artificial neural network which had the highest accuracy of 99.50 per cent and the highest F1-score of 99.52 per cent under all conditions, with the exception that the artificial neural network had the highest performance with the highest accuracy of 99.50 per cent and highest F1-score 99.52 per cent. The addition of the feature extraction process caused only a small increase in computational complexity. The results show that the choice of classifier has an impact on the detection quality, although the differences in competently preprocessed classifiers are not as great as are presented in the literature, and that preparing the data deserves at least the same attention that is usually invested in choosing algorithms. Four directions merit attention. Validation should be repeated on ransomware-specific corpora and on live enterprise traffic, so that the inference from general intrusion detection to ransomware detection is tested rather than assumed. Alternative reduction techniques, including recursive feature elimination and autoencoders, should be compared with principal component analysis under the same controlled design. Hybrid and ensemble configurations that combine the discrimination of connectionist models with the interpretability of tree-based methods deserve investigation, particularly where detections must be explained to non-technical decision makers. Finally, evaluation should be extended to k-fold cross-validation, the Matthews's correlation coefficient, detection latency and memory consumption, which together give a fuller account of whether a model can be deployed rather than merely whether it can classify.

REFERENCES

Akuboh, V. U., Awujoola, O. J., Monsuru, L. A., Shitu, S. S., Adebola, A., Abimuku, C. A., Musa, B. A., & Innocent, E. U. (2026). Enhancing ransomware classification using light weight machine learning algorithm and ensemble methods. FUDMA Journal of Sciences, 10(5), 275–282. https://doi.org/10.33003/fjs-2026-1005-5000

Albin Ahmed, A., Shaahid, A., Alnasser, F., Alfaddagh, S., Binagag, S., & Alqahtani, D. (2023). Android ransomware detection using supervised machine learning techniques based on traffic analysis. *Sensors*, *24*(1), 189. https://doi.org/10.3390/s24010189

Al-Hawawreh, M., Alazab, M., Ferrag, M. A., & Hossain, M. S. (2024). Securing the industrial internet of things against ransomware attacks: A comprehensive analysis of the emerging threat landscape and detection mechanisms. *Journal of Network and Computer Applications*, *223*, 103809. https://doi.org/10.1016/j.jnca.2023.103809

Alraizza, A., & Algarni, A. (2023). Ransomware detection using machine learning: A survey. *Big Data and Cognitive Computing*, *7*(3), 143. https://doi.org/10.3390/bdcc7030143

Al-rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2018). Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Computers & Security*, *74*, 144–166. https://doi.org/10.1016/j.cose.2018.01.001

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

Irankunda, D., El Fazazy, K., Hamid, T., & Riffi, J. (2025). A comparative study of deep learning-based ransomware detection for industrial IoT. *International Journal of Advanced Technology and Engineering Exploration*, *12*(124), 450–466. https://doi.org/10.19101/IJATEE.2024.111101413

Ispahany, J., Islam, M. R., Islam, M. Z., & Khan, M. A. (2024). Ransomware detection using machine learning: A review, research limitations and future directions. *IEEE Access*, *12*, 68785–68813. https://doi.org/10.1109/ACCESS.2024.3397921

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Khan, F., Ncube, C., Ramasamy, L. K., Kadry, S., & Nam, Y. (2020). A digital DNA sequencing engine for ransomware detection using machine learning. *IEEE Access*, *8*, 119710–119719. https://doi.org/10.1109/ACCESS.2020.3003785

Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, *25*(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

Moustafa, N., & Spay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)* (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Oz, H., Aris, A., Levi, A., & Uluagac, A. S. (2022). A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Computing Surveys*, *54*(11s), Article 238. https://doi.org/10.1145/3514229

Serror, M., Hack, S., Henze, M., Schuba, M., & Wehrle, K. (2021). Challenges and opportunities in securing the industrial internet of things. *IEEE Transactions on Industrial Informatics*, *17*(5), 2985–2996. https://doi.org/10.1109/TII.2020.3023507

Urooj, U., Al-rimy, B. A. S., Zainal, A., Ghaleb, F. A., & Rassam, M. A. (2022). Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Applied Sciences*, *12*(1), 172. https://doi.org/10.3390/app12010172

Vehabovic, A., Ghani, N., Bou-Harb, E., Crichigno, J., & Yayimli, A. (2022). Ransomware detection and classification strategies. In *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)* (pp. 316–324). IEEE. https://doi.org/10.1109/BlackSeaCom54372.2022.9858296

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*, 15647. https://doi.org/10.1038/s41598-022-19443-7