International Journal of Research and Development in Engineering Sciences
ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/
Regular Issue, Vol. 8, Issue. 4 (July – August) , 2026 , Pages: 17 - 27
Received: 14 October 2025 ; *Accepted:* 31 July 2026 ; *Published:* 8 August 2026
https://doi.org/10.63328/IJRDES-V8RI4P3
https://www.youtube.com/watch?v=JlQvvO8G44U
https://archive.org/details/jsp-ijrdes-v8ri4p3
Original Paper : Applied Research – Experimental Research Paper

Cybercrime Detection Approaches using Machine Learning and Deep Learning Techniques

Palli Dhilleswari [1] , Jyothi Musireddy [2] , P Vamsi Krishna Raja [3]

Department of Computer Science & Engineering , Pydah College of Engineering , Kakinada , JNTU Kakinada , Andhra Pradesh, India ;
dhilleswaripalli@gmail.com , jyothimusireddy@gmail.com , drpvkraja@gmail.com

* *Corresponding Author : Palli Dhilleswari ; dhilleswaripalli@gmail.com*

Abstract: As a technology that connects even more systems and services to the Internet with each passing day, cybercrime has grown out of control globally. Significant problems with traditional IDSs have been uncovered when they are exposed to new attack vectors and advanced evasion strategies not contained in their rule sets and/or signature databases. This paper contains comprehensive research and empirical analysis of 9 machine learning and deep learning algorithms for binary classification of network traffics into normal or malicious network traffic. The benchmark dataset used in this study is the KDD Cup 1999 dataset, which consists of around 494,021 network connection records derived from a relatively large network, characterized with 41 different features of continuous, discrete and categorical attribute types. Its performance was compared with five classical machine learning algorithms, namely decision tree, random forest, support vector machine with a linear kernel, K-nearest neighbours and gaussian naive bayes. Furthermore, four deep learning architectures were studied, a fully connected Artificial Neural Network, a one dimensional Convolutional Neural Network (CNN), a Long Short-Term Memory (LSTM) recurrent network, and an Autoencoder based model for anomaly detection (AD). These models were thoroughly validated with accuracy, precision, recall, F1 score and ROC-AUC on an 80-20 stratified train test partition. Experimental results showed that the ensemble of Random Forest classifiers has the best overall performance with an overall accuracy of 99.98% and a near-perfect value of ROC-AUC (99.99%). Decision Tree obtained a value of 99.97% in terms of accuracy and a near-perfect value of 99.98% for the ROC-AUC measure. The best deep learning models were the ANN (accuracy of 99.95%) and the LSTM (accuracy of 99.95%) with their nearest architectures, followed by the unsupervised model of Autoencoder (accuracy of 98.94%) with the reconstruction error thresholding. The results highlight that the ensemble tree based methods are still highly effective for the structured network traffic classification cases, and a Deep learning-based approach presents an even more competitive solution with the task of feature selection through raw data, which opens doors of opportunity to extract features by various means from raw data for these cases. In addition, the model was deployed practically with a RESTful API built using FastAPI, allowing for the real-time analysis of live traffic with the trained models.

Keywords: Cybercrime Detection, ML, DL, CNN, LSTM, Autoencoder, KDD Cup 99, Anomaly Detection.

1. Introduction

At the same time, they've made gigantic attack surfaces, which are being used by bad guys more and more with higher technology. In recent estimates, the International Telecommunication Union (ITU) and World Economic Forum (WEF) have estimated that the global impact of cybercrime would be reaching $10.5 trillion annually by 2025 [1] and would become one of the biggest economic threats to organisations and governments all over the world. This growing attack environment calls for more intelligent, adaptive intrusion detection algorithms that are able to run at machine speed and accurately detect known and unknown attack patterns. Intrusion Detection Systems (IDS) is a critical line of Defense in the wider cybersecurity environment. These systems are constantly scrutinizing network traffic and system activities to detect possible issues regarding security, policy or malicious behaviour. IDS implementations have been broadly categorized into two types: implementing signature based detection

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

method and utilizing anomaly based detection method [2]. IDS implementations can be categorized historically into two broad categories: Signature based detection method, which uses a database of attack signature patterns to match observed patterns with known attack patterns and Anomaly based detection method, which establishes a baseline of normal behaviour and flags deviations from this baseline. The signature-based methods work fine against known attacks or variants but introduce serious limitations when it comes to zero day threats or new variants of known attacks. Anomaly-based are systems that might be able to recognize an unknown attack, but often have high false positive rates that can be problematic for the usefulness and operation of a system.

Nowadays, Machine learning and Deep learning techniques are becoming the dominant paradigm in pattern recognition research and this has led to new opportunities for research in intrusion detection. Such data driven approaches can learn the decision boundary directly from labeled training samples, potentially using the accuracy of signature-based approaches combined with the capabilities of anomaly-based approaches to solve a given problem. Some supervised learning algorithms that have shown to perform well on the intrusion detection benchmark datasets are: Decision Trees, Random Forests, Support Vector Machines, K-Nearest Neighbors. In the meantime, deep learning architectures such as feedforward neural networks [13, 14]; convolutional networks [15]; recurrent networks [16] have been promising in the automatic extraction of hierarchical feature representations from raw network data.

Although a mass of literature has been written on this topic, certain difficulties remain. These challenges highlight the complexity of network traffic data, the imbalance in traffic patterns between normal and attack data, the computational burden needed to train the model and execute detection, along with the real-time requirements for detecting the attacks. Also, most of the existing studies report the performance benchmarks of a small number of algorithms only, which is not convenient for model selection [5]. The comparison and evaluation of classical machine learning and the modern deep learning methods under the standardized experimental conditions are still highly desired.

In this paper, these gaps are filled by conducting a systematic review of nine machine learning and deep learning models of binary network intrusion detection and an empirical comparison of the models. The main contributions of this work are fourfold: (i) a comprehensive evaluation of five machine learning and four deep learning algorithms in a standardized manner, both by applying the same preprocessing methods and using the same evaluation metrics on the KDD Cup 1999 benchmark dataset, and by applying them in a real-time system that is deployable for operational usage; (ii) a detailed analysis of model performance using various metrics such as accuracy, precision, recall, F1-score, and ROC-AUC; (iii) a thorough study of training time efficiency as an important deployment consideration for the trained model when operated in real time; and (iv) the development and deployment of a real-time detection system, based on FastAPI, demonstrating the feasibility of the deployment of the trained models for operational usage on network traffic.

2. Related Work

Network intrusion detection using machine learning algorithms has been a research field for more than 20 years, and has produced many key papers that have been reinforced with the emergence of benchmark datasets like the DARPA 1998 evaluation data set and KDD Cup 1999 data set. Subsequent researchers have heavily expanded upon an early feasibility study by Lee and Stolfo [6] which showed that decision tree classifiers as well as association rule mining can be used to detect anomalous network connections. Mukherjee and Sharma [7] carried out a thorough study of the machine learning techniques used in IDS, which they divided into two types: use detection and anomaly detection. The analysis showed ensemble methods, such as Random Forests and boosted tree models, generally surpassed individual classifiers when tested on several benchmark sets. In the context of cybersecurity intrusion detection, feature selection and dimensionality reduction techniques, both of which are a critical part of the process of achieving an effective classification rate, were also revisited [8] in relation to data mining and/or machine learning algorithms. With the recent rise of deep learning, architectures have entered the scene that were able to learn feature representations hierarchically without any manual feature engineering. Non-symmetric deep autoencoder for intrusion detection (Shone et al. [9]): They showed that by unsupervised pre-training they could outperform other methods on the KDD Cup 99 and NSL-KDD datasets. Their ability to detect anomalies using the model's reconstruction error scores, when the attack causes an elevated error score, were useful indicators of the potential for autoencoders to detect anomalies.

Kim et al. [10] studied the use of LSM network for intrusion detection, considering network traffic as a time series. The experiments which they conducted on the KDD Cup 99 dataset proved that the LSTM architectures could be used to apply sequential dependency to network connection records and it was able to detect cases with rate more than 98% accuracy. Later, the study was expanded by looking at different types of deep-learning-based architectures such as CNN, LSTM and hybrid CNN-LSTM model by Vinayakumar et al. [11] and it has been found that CNN-

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

based approaches are quite effective in extracting spatial features from the representation of network traffic.

More recent investigations have looked at the viability of the implementation of ML-based IDS. In real-world deployment, Khraisat et al. [12] noted that the factors of latency requirement and throughput constraint are important. Ahmad et al. [13] suggested a system of two stages: first, feature selection and second, ensemble classification in conjunction with realtime networking intrusion detection system (IDS). These works highlight not only that the classification accuracy is essential, but also the need to think of efficiency and capability for deployment when designing an IDS. The current studies offer some insights, yet there are several reports that test just a few algorithms or take a look at only whether it is machine learning or deep learning strategies. This paper fills this void by performing a coherent comparative study with a similar data-preprocessing, evaluation score and evaluation protocol that is consistent across paradigms and allows fair and meaningful comparisons to be drawn.

3. Dataset and Preprocessing Methodology

In this paper, the authors employ the KDD Cup 99 data set for the experimental evaluation which is one of the most frequently used data sets when evaluating a network intrusion detection system. The data was obtained from the DARPA Intrusion Detection Evaluation program held at the MIT Lincoln Laboratory in 1998 [14] where a simulated military scenario was attacked by many different attacks for nine weeks. This raw tcpdump information was then post-processed by Stolfo et al. to produce significant features at the connection level, which was used as the data set to be released for the data mining event KDD Cup 1999.

The fact that each row of the dataset specifies a single network connection with its 41 features, which describe various aspects of the connection usage, made it easy to find the right way to use it in our work. The ease of use was due to the fact that every record in the data set describes one network connection and each of its characteristics is recorded in one of the 41 features. The features have been sorted into three concept clusters. The first category includes basic data about individual TCP connections, such as connection length, protocol type, service, statuses about flags, source and destination bytes of connection, and different error indicators. The second form involves domain-specific information features, such as log-in failures, successful log-in to a root shell and indicators of suspicious file operations such as those related to attack types. The third category encompasses time-based and host-based traffic properties measured over a temporal window of two seconds that includes characteristics useful for detecting scanning and denial of service patterns such as connection rates, service error rates and other statistics derived from traffic.

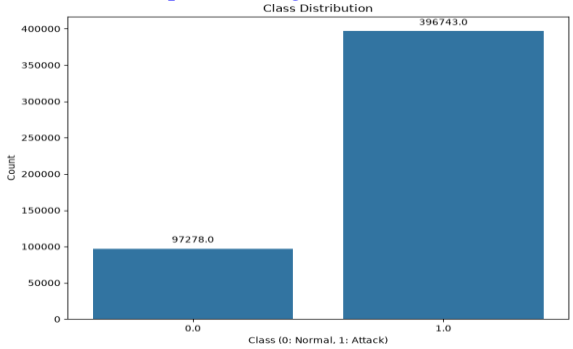


Fig. 1 Class distribution of the KDD Cup 99 dataset showing the proportion of Normal and Attack instances.

The data consists of five main categories: Normal connections, Denial of Service (DoS), Remote-to-Local (R2L), User-to-Root (U2R) and Probing attacks. These attacks are examples of DoS attacks that occur in the data set and can include attacks such as smurf, eptune, back, teardrop, and land. Some types of R2L attacks include spy, warezclient, warezmaster, and ftp_write, etc. The buffer_overflow, rootkit, loadmodule and perl exploits represent U2R attacks. The probing attacks involve the scanning activities such as nmap, port sweep, ip sweep and satan scan [15]. In this study, all the multi-class labels were reduced into a binary classification scheme of Attack class (deduced from all attack types) equals label 1, and Normal class equals label 0. It is a binary formulation that allows simplifying the detection job into distinguishing between benign (good) and malicious (bad) traffic - the most basic and operationally relevant classification in network security. It was decided to use a subset of the overall data, 10%, which contained around 494,021 connection records and enabled a training and testing sample of sufficient size and representative of the overall data set, but still allowed for the required level of computational processing.

3.1. Data Preprocessing Pipeline

The data was pre-processed using a systematic pipeline to ensure that the raw data is ready for use in machine learning and deep learning models. The pipeline has multiple stages, each of which is trying to solve a particular issue related to data quality or representation. The first was loading it using all the \%10 or these sorts of parameters in the scikit-learn fetch_kddcup99 so that the 10% of the dataset was downloaded. All the string columns were translated from a byte-encoded representation, to a standard UTF-8 string representation, to ensure uniformity of processing down-stream. Treatment of missing values involved removing rows with any null or NaN values from the dataset with the pandas dropna function. The KDD Cup 99 dataset has not many problems in terms of missing data and is in general a good dataset, but this ensures that the preprocessing pipeline is robust when using potentially noisy real world data.

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

The data set was found to have three categories: protocol_type (tcp, udp, icmp), service (for the purpose of this data set, the service is considered to be a network service to which a connection is made such as http, smtp, or ftp), and flag (the state of the connection, for this data set, either accept or reject). All these categorical variables were transformed by one-hot coding using the pandas get_dummies function with drop_first set to True to prevent multicollinearity. This is the encoding when the categorical value is being encoded by creating a binary indicator variable for each categorical value, except for the first value in each category which is dropped as the reference class. All numerical attributes have been scaled by the StandardScaler from scikit-learn, to have zero mean and unit variance. This step of normalisation is especially relevant to distance-based measurements like KNN and SVM, and can also help to make the gradient descent process in deep learning models converge quicker. The scaled data-set is then divided into Random sampling to achieve an equal distribution of each class in the training and testing subsets (80/20). Throughout, fixed random state 42 was used to promote replicability.

3.2. Machine Learning Models

In the supervised classification paradigm, five classical machine learning algorithms were chosen that highlight the various concepts of learning. The criteria considered were aligned with a desire to promote fundamentally different inductive biases, learning mechanisms, and computational properties; to allow for a full analysis of the algorithms, they were selected. The Decision Tree classifier uses a recursive partitioning approach to build a recursive decision tree as a decision structure where at each node the best feature and threshold to choose is the one that provides the greatest information gain or the lowest Gini impurity. The implementation calls the CART algorithm that is simply seeded at random state 42. Although Decision Trees are relatively easy to train and understand, they can suffer from overfitting when presented with large, complex datasets and can be applied in many situations where there are numerical and categorical features, without intensive feature engineering. It is an extension of decision tree, where multiple decision trees are built in the learning process, and voting is performed using the majority of the classifiers. An ensemble of 100 individual trees was used in this study (bootstrap sampling, random feature selection at each split). It is believed that the Random Forest generates more reliable and generalizable models with lesser variance associated with the individual trees and at the same time remains efficient on computation.

In fact, the class of maximum margin classifiers was represented with the Support Vector Machine with a linear kernel. SVM aim at locating an optimal separable hyperplane which maximizes the geometric distance of the classes from one another in the feature space. The kernel used is linear and the maximum number of iterations is 1000. Unlike LAsVMs, SVMs are soundly theoretical and work well in high dimensions, but may have scalability issues with very large data sets. The K-Nearest Neighbors algorithm is one of the various families of instance-based learning algorithms that put computation in the prediction stage. For every test instance it finds the K closest examples in the feature space using Euclidean distance and labels them as per the top most class amongst the closest examples. The parameter K was also set to 5, which is a well-known parameter to optimize bias and variance trade-off. KNN is a non parametric approach as it does not assume distribution of underlying data, thereby giving flexibility but may become expensive at prediction time in case of larger datasets. The Gaussian Naïve Bayes classifier uses the above Bayesian approach with the simplifying assumption that the features are conditionally independent given the class label. The simplifying assumption is rarely implemented in practice, but Naïve Bayes classifiers can nevertheless turn out to be very accurate especially when the number of D's is large compared to the number of training elements. A major advantage of this probabilistic classifier is that only one single pass through the data is needed to estimate class conditional feature distribution with small complexity. The merit of this probabilistic classier is the training time complexity which is small, as only one single pass through the data is needed to estimate class conditional feature distribution with small complexity.

3.3. Deep Learning Models

This paper proposes the design and evaluation of four deep learning architectures to understand the neural network model capability in network intrusion detection. The code for all deep learning models was developed in-Aug 2019.The deep learning models all developed Aug 2019. The models were trained using the Adam optimization algorithm, with a Binary Cross Entropy (BCE) loss function (mean squared error (MSE) loss function for autoencoder). The amount of training that was done was 10 epochs, each with 64 batch size with 10% of the data used as a validation set to ensure that progress is monitored during training. The Artificial Neural Network (ANN) architecture is fully connected feed forward network with three hidden layers having 128, 64 and 32 neurons respectively with Rectified Linear Unit (ReLU) Activations. To prevent overfitting, dropout is used at a rate of 0.3 after the first two layers of the hidden layers. The final layer (output) has one node with sigmoid activation to output the probabilities for the two states of the binary classification. This architecture aimed to achieve a trade-off between the capacity of the model and the regularization of the network, promoting the learning of non-linear decision boundaries and preserving the generalization of the model. Convolutional Neural Network (CNN) was successfully adapted to the tabular network traffic data by considering

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

each feature as a position of a one dimensional sequence.

The architecture is made up of 64 filters with kernel size 3 and ReLU activation in the Conv1D layer and MaxPooling1D layer with a pool size of 2. The convolutional features are flattened and then fed through a dense layer of 64 neurons with dropout regularisation (rate 0.3) before sigmoid output layer. Data was reshaped into the format (samples, features, 1) to generate and analyze a one-dimensional representation of a single channel. This architecture takes advantage of the fact that the convolution operation is able to capture local patterns and feature interactions in the ordered feature space. A sequential modeling architecture was examined using so-called Long Short-Term Memory (LSTM) network to test if the temporal or ordinal information contained in the feature vector is meaningful. The architecture is a single LSTM layer of 64 neurons followed by a dropout regularization layer (0.3), a dense layer with 32 neurons and a output layer of sigmoid. This input was reshape into (samples, 1, features) where each one of the connection record represents a time step with multiple features. The network traffic is sequential at the packet level though, while the connection-level features in the KDD dataset are aggregated network statistics; however, it is likely that the interactions of complex features are preserved in the LSTM's gating mechanisms which will aid in classification.

3.4. Evaluation Metrics

Various evaluations metrics were used to evaluate the models from many different points of view. Overall measure of classification correctness is given by accuracy which measures the proportion of the correct classification cases to the total cases. The accuracy alone, though, is not a meaningful measure in the context of class imbalance, however, so other measures were computed for a more complete evaluation. The proportion of instances predicted as attacks as malicious: a measure of the model's ability to avoid false alarms, also known as Precision. In operational environments, you want to be as precise as you can to avoid alert fatigue by security analysts. Recall (sensitivity or true positive rate) is the percentage of individuals who suffered a true attack who were classified as having an attack. In the security space, it's probably the most important measure; with missed attacks (false negatives) having dire implications. F1 score is the harmonic average of precision and recall, a single "balanced" score that considers both missed positive and false positive results. This metric is specially helpful for analyzing models on an imbalanced dataset. The Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) presents the discriminative power of the model over all possible classification thresholds, with a value from 0.5 (random guessing) to 1 (perfect discrimination). Furthermore, confusion matrices were created for each model to see the distribution of true positives, true negatives, false positives and false negatives.

4. Experimental Results And Analysis

This section presents the experimental outcomes obtained from training and evaluating all nine models on the KDD Cup 99 dataset. Table I provides a comprehensive summary of performance metrics for each model, while the subsequent subsections offer detailed analysis of the results across different model categories.

Table.1 Comparative Performance of All Models on KDD Cup 99 Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)	Training Time(s)
Decision Tree	99.98	99.98	99.99	99.99	99.96	8.21
Random Forest	**99.98**	**99.99**	**99.98**	**99.99**	**100.00**	**65.67**
SVM	81.08	81.28	99.31	89.39	96.32	143.72
KNN	99.95	99.97	99.96	99.97	99.97	0.63
Naïve Bayes	94.95	99.97	93.74	96.75	97.76	2.71
ANN	99.95	99.97	99.96	99.97	100.00	342.20
CNN	99.94	99.96	99.96	99.96	100.00	465.14
LSTM	99.95	99.98	99.95	99.97	100.00	249.14
Autoencoder	98.94	98.72	99.97	99.34	99.19	40.52

4.1. Machine Learning Model Performance

The five evaluation machine learning algorithms, the tree-based ensemble methods outperformed the rest of the classifiers in terms of classification. The accuracy of the Random Forest classifier was the highest at 99.98% and the value of precision, recall, and F1-score were all greater than 99.98%. The ROC-AUC value is near to 1, showing excellent discriminative power for each threshold of classification. The accuracy achieved by the Decision Tree was 99.97% and the F1-score was 99.98%, which is close to the result obtained by the Inference Tree. This is as expected and successful in the well-known context of tree-based methods that work well with structured tabular data, with complex feature interactions can be modelled very well using recursive partitioning. This conclusion is supported by the fact that K-Nearest Neighbours with balanced precision and recall accuracy of 99.95% well classifies by the representations of feature space were scaled with standard scaling, which proves that feature space representation after standard scaling is enough to be separated for distance-based classification. Moreover, KNN recorded the minimum time taken in training, which was 0.63 seconds, out of all the models because it does not compute an explicit model during its training.

The Gaussian Naïve Bayes classifier was able to gain 94.95% accuracy, which was good, but still fell behind the tree-based and instance-based classifiers. This is mainly because of the presence of a strong conditional independence assumption between network traffic features that are correlated to each other, which is likely not met. For Naïve Bayes, the precision (99.97%) is very high,

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

meaning that, although the recall (93.74%) is low, its predictions are very reliable for positive values. The linear SVM model had the least accuracy out of all models, with 81.08% accuracy. This outcome is due to the fact that the class separation is nonlinear in the feature space, but the learning algorithm used is limited to a linear Entscheidungsgrenze. Other possible reasons for potential lack of convergence were the size and dimensionality of the data set, as well as the maximum number of iterations allowed of 1000. Although the accuracy is low, the SVM showed 99.31% recall, which indicates it is more of an aggressive classifier and gives more false positives which adversely affects the overall precision.

4.2. Deep Learning Model Performance

Overall the performance of the deep learning models was good, with accuracy of ANN, CNN, and LSTM models ranging from 99.94% to 99.95%. The accuracy of the ANN was 99.95%, that of the LSTM was 99.95% and for the CNN it was 99.94%. The results show that these deep learning models perform on par with the best machine learning models and are capable of learning complex decision boundaries by means of several layers of nonlinear transformations. The ANN architecture, with its three hidden layers, dropout regularization and its complexity, was found to have near-optimal performances for this classification task. This gradual shrinking of the layer size from 128 → 64 → 32 neurons puts an "information bottleneck" on the network making it learn increasingly abstract feature representations. The training time of 342.20 seconds accounts for the computational burden of gradient-based optimization over a number of epochs, significantly higher than that of the tree-based machine learning approaches. The 1D CNN showed a 99.94 accuracy which shows the effectiveness of the convolutional operations to learn local feature patterns from tabular data when features are organized as a one dimensional sequence.

The convolutional kernels of size 3 enable the network to learn the relationship between neighbouring features, which can be semantically similar network connection attributes. The training time for 465.14 seconds was however the most for all models, due to the extra computational cost of the convolution operations. For the best LSTM network; the accuracy was 99.95%, and training time was 249.14 sec. The features in KDD do not form a natural time sequence, however, the intricate gating mechanisms of the LSTM seems to have successfully captured complex feature dependencies. In the case of KDD the features do not represent a natural evolution over time but nevertheless the LSTM's more involved gating processes show an ability to model complex feature dependencies. Each LSTM cell features a forget, an input and an output gate, allowing the network to selectively remember or forget information, and potentially capturing subtle features interactions that are relevant to a correct classification.

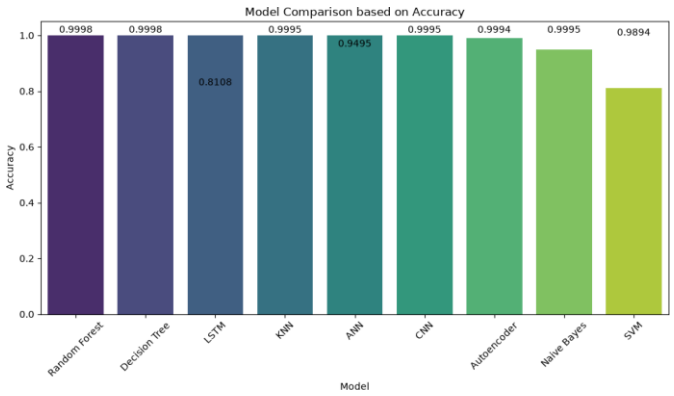


Fig. 2. Comparative accuracy of all nine models evaluated on the KDD Cup 99 test set.

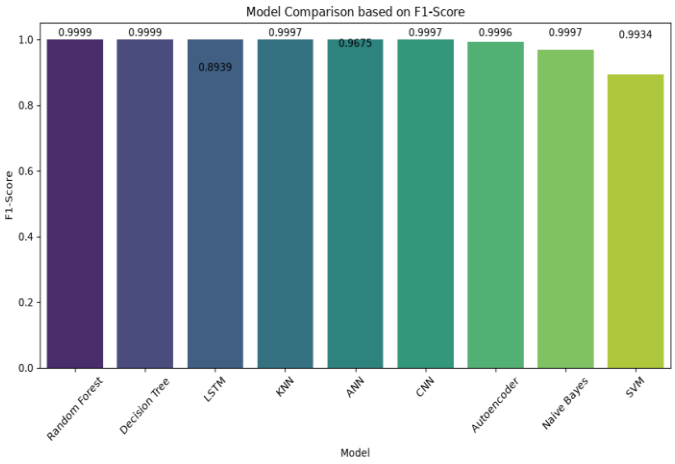


Fig. 3 F1-score comparison across machine learning and deep learning models.

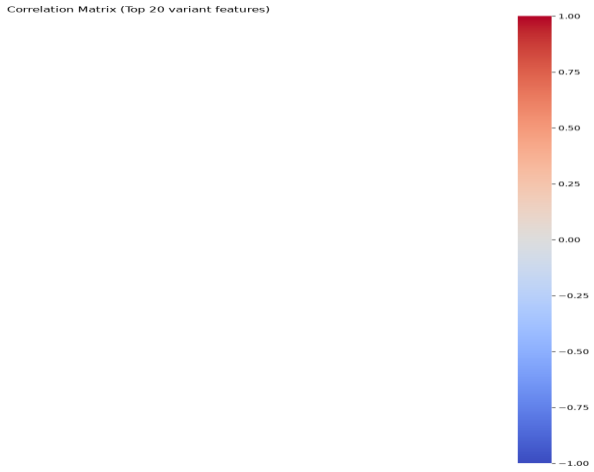


Fig. 4 Pearson correlation matrix of the top 20 numerical features in the KDD Cup 99 dataset.

The overall performance is still good, although not the best among deep learning architectures, with the accuracy of the Autoencoder model being 98.94%. First, it needs to be noted that the Autoencoder is an unsupervised anomaly detector; the algorithm learns from only normal traffic samples, and has not been trained with knowledge of

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

attack patterns; it is different than supervised anomaly detection. The high recall of 99.97% means that it has excellent sensitivity toward detecting anomalous connections, and its relatively lower precision of 98.72% signposts that some normal connections having unusual feature patterns are miss-classed as attacks. The threshold-based classification approach based on 95% of the following data is applied. This is applied with the 95th percentile of the data applied to the threshold based approach.

4.3. Confusion Matrix Analysis

In Figures 5 to 13 detailed explanations of the classification behaviour of each model are presented in the form of confusion matrices. The pattern of diagonal elements dominate in both confusion matrices of the tree-based models and deep learning architectures, which means that for most of the cases the test results (in normal and attack classes) are all on the same diagonal.

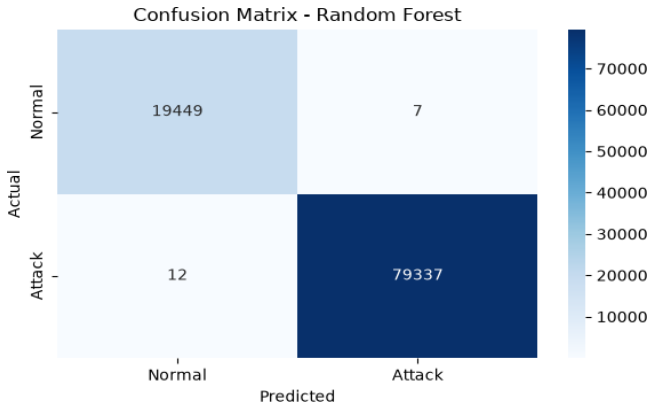


Fig. 5 Confusion matrix for the Random Forest model.

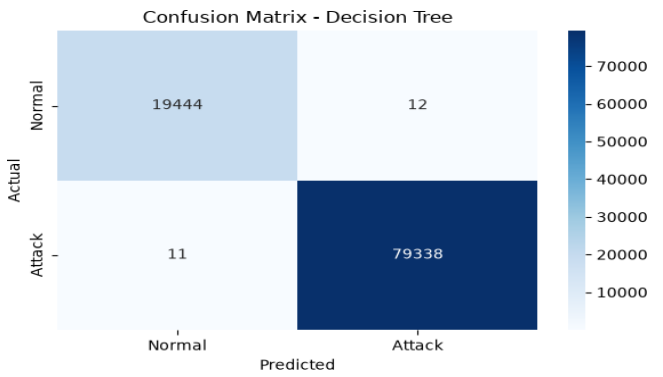


Fig. 6 Confusion matrix for the Decision Tree model.

The Random Forest and Decision Tree algorithms correctly classify less than 25 rows for the test set with more than 98,000 rows. The SVM-confusion matrix confirms that the model is prone to classify threats as aggressive attacks, and there is a considerable amount of normal connections misclassified as threats. As per the low recall value, there is a definite cluster of false negatives in the Naïve Bayes model. A supervised model, such as the ANFs or LR, controls the error granularity, whereas the unsupervised Autoencoder's confusion matrix shows the slight broader distribution of errors and yet it benefits from a very impressive detectability.

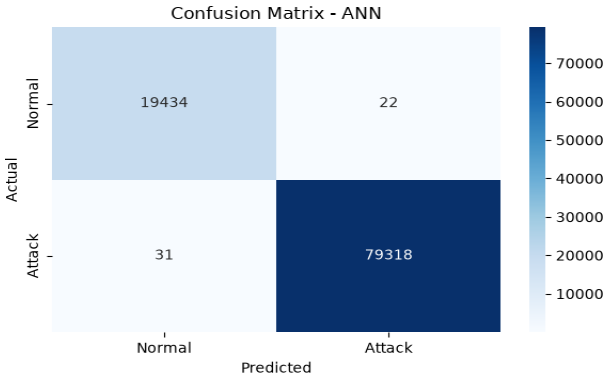


Fig. 7 Confusion matrix for the Artificial Neural Network model.

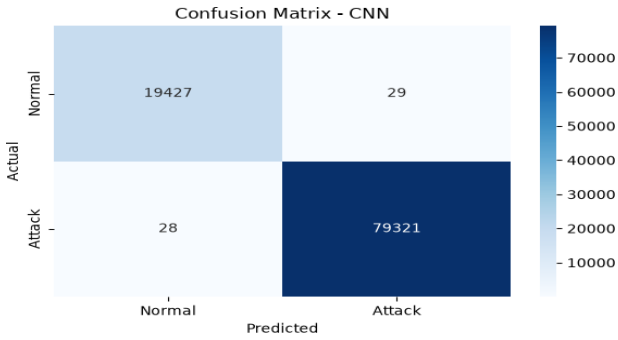


Fig. 8 Confusion matrix for the Convolutional Neural Network model.

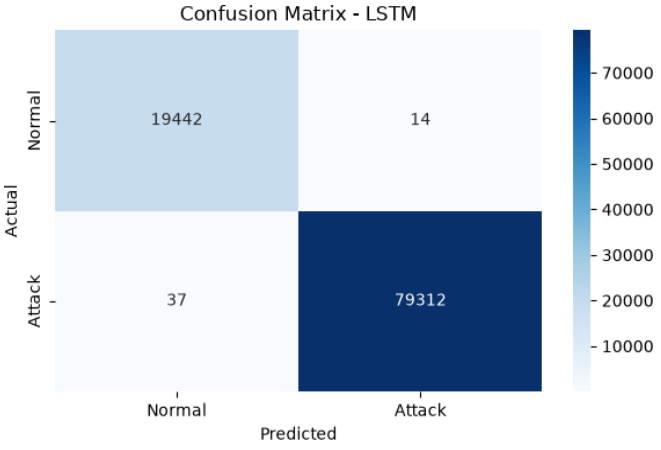


Fig. 9 Confusion matrix for the LSTM model.

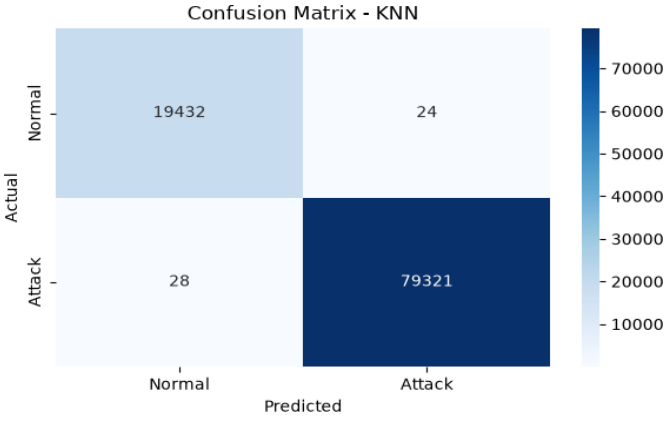


Fig. 10 Confusion matrix for the K-Nearest Neighbors model.

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

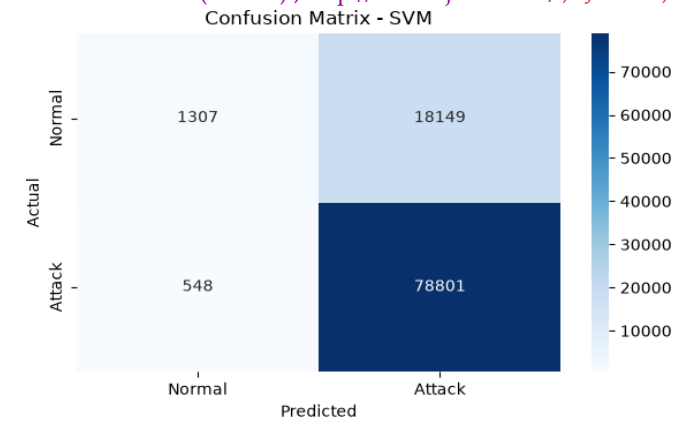


Fig. 11 Confusion matrix for the Support Vector Machine model.

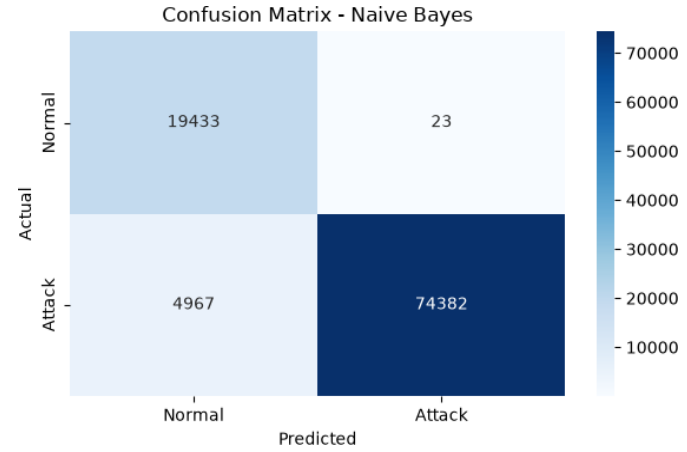


Fig. 12 Confusion matrix for the Naïve Bayes model.

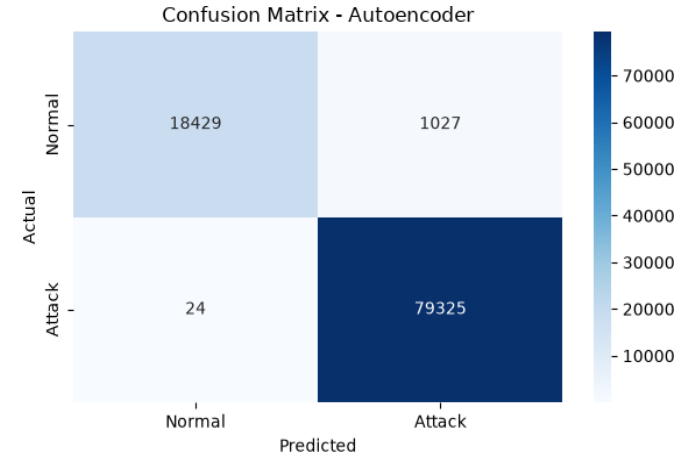


Fig. 13 Confusion matrix for the Autoencoder model.

4.4. Training Time Efficiency Analysis

Apart from the practical deployment considerations, training time is a key aspect of deploying intrusion detection systems, especially when the system is expected to be updated from time to time to suit changing threat landscapes. KNN had the lowest nominal training time when compared with other machine learning models and had the highest efficiency among them. The above number is 0.63 seconds, which is not an ideal value and also the computation cost of KNN is deferred till prediction time. The Decision Tree needs 0.0821 seconds, whereas the Random Forest (100 Individual Trees) needs 0.0657 seconds to be trained. The Gaussian Naïve Bayes classifier was trained on These results are achieved in 2.71 seconds, which represents the one-trip estimation of class-conditional distributions of the algorithm. The most time-consuming ML model in terms of training times was the Linear SVM at 143.72 seconds of training time because of the iterative optimization to ensure the maximum-margin hyperplane.

The deep learning models had significantly higher training times: 40.52 seconds for the Autoencoder, and 465.14 seconds for the CNN. The number of these training time corresponds to the computation cost of 10 training epochs using backpropagation-based gradient optimization. The ANN needed 342.20 seconds to perform training and the LSTM needed 249.14 seconds to perform training. The advantage of deep learning is marginal, where the accuracy estimated is less than 0.05% when compared to the accuracy achieved by the Random Forest model, and the computation time required for training a deep learning model in this context can be substantial, so deep learning may have some advantages if the data to be classified is raw packet-level data or if multiple classes of attacks are to be categorized.

5. Real-Time Detection System Architecture

A practical demonstration of the trained models was done by creating a real-time Intrusion detection system (IDS) using the FastAPI web framework.To prove the practical viability of the trained models, a real-time intrusion detection system (IDS) was created, based on the FastAPI web framework. It is designed as a client server model, and the best model (Random Forest) and the entire preprocessing process is serialized by joblib and loaded at server startup. There are a number of RESTful API endpoints available for who can interact with the models on the server. The primary prediction endpoint can receive one or more feature vectors in the form of JSON, then perform the same feature pre-processing steps against the same data as those performed in the training phase (encoding and scaling), and will return not only the classification result, but also the probability associated with each class. There are some other endpoints added that return precomputed model comparison results, sample normal and attack feature vectors to be tested, and model performance metadata.

In order to provide a user friendly interface to the detection system, an interactive Web-based dashboard was developed. The dashboard has a glass morphism design approach and dark theme and has several functional panels, such as model performance leaderboard, live prediction testing panel, custom payload upload panel, and an automated, real-time streaming prediction test mode which generates and classify synthetic traffic samples at a predetermined frequency. Additionally, a module has been

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

added for performing the analysis of traffic at the domain level, that is built via the Scapy packet manipulation library. It allows targeting a domain, automatic domain to ip address translation, generating feature vectors as KDD-99 compatible vectors from the captured network traffic after a configurable period of time; and can be used for real-time classification. This ability shows the possibility of adapting the trained models for use in real network monitoring processes.

5.1. DISCUSSION

The results of the experiments show several interesting points of interest which are discussed below. That tree-based ensemble models, such as Random Forest, outperform other models according to the results confirms results from previous research and the relevance of classical machine learning models for structured classification. This high accuracy of the Random Forest with the balanced precision and recall shows that (in the specific feature space) we do not have to use complex deep learning architectures to got state of the art performance. This observation has some applications because tree-based models have several advantages over neural network options for both interpretability, ease of training and ease of deployment. It is important to note that all the architectures of ANN, CNN, and LSTM models perform comparably with an accuracy margin ranging within 0.04%, indicating that model selection is not a major consideration for this specific dataset when using the neural network approach, as the performance of the network is mostly dependent on the quality of feature representation.

6. Conclusion and Future Scope

In this paper, a thorough review and empirical comparison of nine of the machine learning and deep learning models employed in the network intrusion detection for benchmark dataset of KDD cup 1999 are provided. The consistency of the experimental conditions with which five classical ML algorithms (KNN, Decision Tree, Random Forest, SVM, Naïve Bayes) and four architectures with DL (ANN, 1D CNN, LSTM, Autoencoder) have been evaluated allows for an easy comparison between the different paradigm classes. Experimental results show that the Random Forest ensemble classifier has the best overall performance with an accuracy of 99.98%, precision of 99.99%, recall of 99.98% and almost perfect ROC-AUC score. The accuracy of deep learning models ranged from very competitive (ANN, LSTM at 98.94% to 99.95%) to competitive (ANN, LSTM at 98.94% to 99.95% which were competitive with the second best of machine learning model (Decision Tree at 99.97%). Even though the Autoencoder system works without supervision, it gets 98.94% accuracy, presenting the ability of reconstruction-based anomaly detection to be used in network security applications. Practical implementation of the optimal model has been achieved with the implementation of a REST API based on FastAPI and interactive web dashboard, to prove that research models can be readily put into operation as detection tools. With the added functionality of a live packet sniffer, the utility of the system greatly enhances when used in real live network monitoring scenarios.

The future directions for research involve evaluating the methods on more modern examples of network traffic like the CICIDS 2107/2018 and the UNSW-NB15 dataset, which captures the network traffic flow types and attack patterns prevalent in modern networks. If multi-class attack could be extended to it than it can be categorized based on threats with a finer granularity. Modeling interactions between features could be enhanced through exploration of architectures built with attention such as Transformers variants with Tabulateer inputs. Models of attention like Transformers variants with Tabulateer inputs will bring an improvement in feature interaction Modeling. Furthermore, federated learning solutions for privacy-preserving multi-network domain collaborative model training algorithms are an intriguing factor to consider for deployability with scale. Lastly, an adversarial robustness test should be performed to assess the robustness of the model to evasion attacks that are intentionally designed to fool the machine learning models used to detect attacks.

References

[1]. S. Morgan, “Cybercrime to cost the world $10.5 trillion annually by 2025,” Cybersecurity Ventures, Nov. 2020. [Online]. Available: https://cybersecurityventures.com

[2]. H. Liao, C. Lin, Y. Lin, and K. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16-24, 2013.

[3]. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, 2016.

[4]. M Anjan Kumar , P Viswanatha Reddy , T Gopikrishna , MD Akbar , “ Real-Time Facial Emotion Recognition Using Deep CNN and YOLO-Based Face Detection: A Hybrid CNN–LSTM Framework “ , International Journal of Computer Science , Engineering and Artificial Intelligence , Vol. 2, Issue. 2 (April – June) , 2025 , Pages: 14 - 23.

[5]. Suresh Kodeboina, Akbar Sk, andT Murali Mohan , “Spatio-Temporal modeling for abnormal behaviour detection in crowd scenes,” *International Journal of Computational Science and Engineering Research*, vol. 3, no. 1, p. 37, Jan. 2026, doi: 10.63328/ijcser-v3ri1p4.

[6]. V. B. Reddy, B. Reddy, N. Shanthi, I. A. M. S, and D. A, “Fake job recruitment detection using machine learning,” *International Journal of Computational Science and Engineering Research*, vol. 2, no. 3, p. 9, Aug. 2025, doi: 10.63328/ijcser-v2i3p2.

[7]. M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, p. 102419, 2020.

[8]. N. Moustafa and J. Slay, "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, vol. 25, no. 1-3, pp. 18-31, 2016.

[9]. J V G Prakasa Rao Pyla , “ Transformer-Based Real-Time Wing Flutter Detection in Formula 1 Vehicles “, International Journal of Computer Science , Engineering and Artificial Intelligence , Vol. 2,

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3

Issue. 1 (January – March) , 2025 , Pages: 18 – 24.

[10]. W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 227-261, 2000.

[11]. S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," Procedia Technology, vol. 4, pp. 119-128, 2012.

[12]. M. K. C and L. Gudivada, "A Hybrid Deep Learning Approach for Early Parkinson's Detection from Handwriting," *International Journal of Research and Development in Engineering Sciences*, vol. 7, no. 4, p. 1, Jul. 2025, doi: 10.63328/ijrdes-v7ri4p1.

[13]. M. S. S, A. R. T, V. P, and N. Gandla, "An Attention-Enhanced YOLOV8 model for accurate Multi-Class kidney abnormality detection in CT imaging," *International Journal of Computational Science and Engineering Research*, vol. 3, no. 1, p. 68, Jan. 2026, doi: 10.63328/ijcser-v3ri1p8.

[14]. M Anjan Kumar , P Viswanatha Reddy , T Gopikrishna , MD Akbar , " Real-Time Facial Emotion Recognition Using Deep CNN and YOLO-Based Face Detection: A Hybrid CNN–LSTM Framework " , International Journal of Computer Science , Engineering and Artificial Intelligence , Vol. 2, Issue. 2 (April – June) , 2025 , Pages: 14 - 23.

[15]. Bharat Kumar Chigilipalli , VenkataRamana Guntreddi , " Multi - Modal Driver Fatigue Detection Using Computer Vision and Biometric Sensor Fusion ", International Journal of Computer Science, Engineering and Artificial Intelligence , Vol. 3, Issue. 1 (January – March), Pages: 19 – 26 , 2026

How To Cite:

Palli Dhilleswari , Jyothi Musireddy , P Vamsi Krishna Raja, Cybercrime Detection Approaches using Machine Learning and Deep Learning Techniques , International Journal of Research and Development in Engineering Sciences (IJRDES), Volume 8, Issue 4 , pp 17 - 27 , August , 2026.
CrossRef DOI: https://doi.org/10.63328/ijrdes-v8ri4p3

Declaration

Conflicts of Interest: The authors declare no conflict of interest.

Author Contribution: All authors wrote the main manuscript text and also consent to the submission.

Ethical approval: Not applicable.

Consent to Participate: All authors consent to participate.

Funding: Not applicable, and No funding was received

Institutional Review Board Statement: Not applicable.

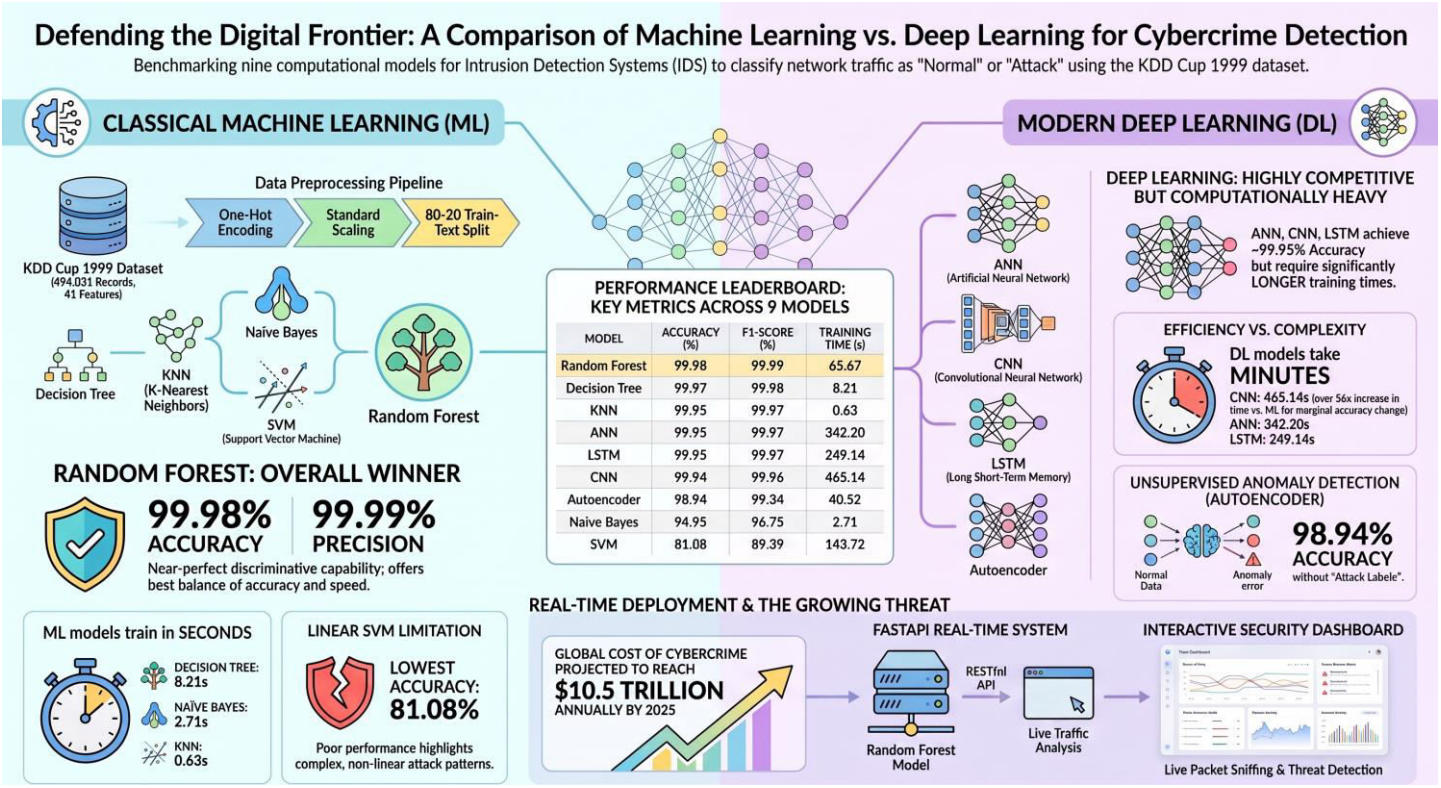
Informed Consent Statement: Not applicable.

Personal Statement: We declare with our best of knowledge that this research work is purely Original Work and No third party material used in this article drafting. If any such kind material found in further online publication, we are responsible only for any judicial and copyright issues.

Acknowledgements

We thank everyone who inspired our work.

: : Overview of the Paper for Researchers : :



ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 8, Issue 4 , 2026 , https://doi.org/10.63328/IJRDES-V8RI3P3