Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 2 (2025)

# Advanced Deep Learning Techniques for Information Security Vulnerability Detection Using Machine Learning

**Champa Tanga[1], Madhukar Mulpuri[2], Dr.A. Mahalakshmi[3], P.K. Hemalatha[4], Dr. Gurwinder Singh[5], Tareek Pattewar[6], Nargis Parveen[7]**

[1] Assistant Professor, Department: Electronics and and Communications Engineering Rajiv Gandhi University, Arunachal Pradesh. Champa.tanga@gmail.com

[2]Senior Staff Engineer, NIUM Inc, connectmadhukar@gmail.com

[3]Associate Professor, Department of Management Studies, M S Ramaiah Institute of Technology, Bangalore, mahalakshmi.a@msrit.edu

[4]Assistant professor, Dept of Mathematics, Vel Tech Rangarajan Dr Sagunthala R & D Institute of Science and Technology, Chennai , India, pkhemalathamsc@gmail.com

[5]Associate professor, Department of AIT-CSE Chandigarh University, Gharuan, Punjab, India
Email: singh1001maths@gmail.com

[6]Assistant Professor, Department of Computer Engineering, Vishwakarma University Pune, tareek.pattewar@vupune.ac.in

[7]Department of Computer Science, Faculty of Computing and Information Technology, Northern Border University, Kingdom of Saudi Arabia, nargis.norulhaq@nbu.edu.sa

***Article History:***

***Received:** 18-07-2024*

***Revised:** 08-09-2024*

***Accepted:** 29-09-2024*

**Abstract:**

The increasing rate on the complexity and amount of security threats demand more advanced information security vulnerability detection techniques. Traditional methods have limited capability to respond to such diverse and evolving threats in real-time. This paper presents a complete solution by using new machine learning models in combine with advanced mathematical approaches to organize the early detection and prediction of vulnerabilities that later content will be about the information security systems. We investigate the accuracy of different models (with an emphasis on modern approaches that go beyond the standard SVM, CNN, and GNN). It initiates with Bayesian Networks that use probabilistic graphical models for showing the interactions of diverse security characteristics to support reasoning about vulnerabilities as a function on the conditional dependencies. Decision Trees and ensemble techniques like Extreme Gradient Boosting (XGBoost) are able to deal with both heterogeneous data types and are more capable of modelling complex interaction between security features. For unsupervised cases, we use things like Isolation Forests for anomaly detection and Gaussian Mixture Models (GMM) to detect rare patterns in network packets that may be a sign of security attacks. We further investigate the possibility of reinforcement learning (i.e., Q-Learning) to continue to evolve with network changes and identify threats in real-world threatening environment. The learning process is streamlined and the interpretability of results is improved by integrating mathematical techniques such as Markov Chains (for modelling probabilistic transitions within network states) and optimization methods like Lasso Regression (for feature selection). In addition to this, the approach explores Autoencoder-based models but most specifically Variational Autoencoders (VAEs) for their unsupervised learning ability in identifying rare and new zero-day vulnerabilities. Extensive experiments on a variety of cybersecurity datasets demonstrate the effectiveness and efficiency of our approach, where substantial enhancements in detection speed and accuracy are achieved. The cloud-based cybersecurity model introduced in this research, backed by advanced machine learning models and mathematical frameworks could potentially serve to lay a foundation for the future of real-time cyber security defence mechanisms.

**Keywords:** Complexity, decision, unsupervised, detection, datasets, security, network, mechanisms.

## 1. Introduction:

Traditional security mechanisms are being put to the test as cyber threats increase in complexity and volume, making it much harder for them to detect vulnerabilities effectively. The basic foundation of Cyber Security is to identify and remediate the security vulnerabilities before it could be used against an organization. Traditional methods like signatures and rules are often ill-equipped to identify new or complicated attacks, thus necessitating more advanced means of detection. In machine learning, over the past few years this concept of using machine learning with math techniques has gained a lot of traction and also proven itself as an effective way to improve your vulnerability detection. In this paper, we probe deeper into how state-of-the-art machine learning models can complement sophisticated mathematical frameworks in the realm of information security vulnerability detection[1].

● Mathematical Approaches to Vulnerability Detection:

Mathematics is the backbone of many algorithms and methods in machine learning and data analysis. Mathematical methods play critical roles in information security, including data pre-processing, feature extraction, dimension reduction, anomaly detection, and model optimization. Important Mathematical Methods Applied in Cybersecurity[2].

○ Linear Algebra and Matrix Factorization: Commonly applied for the purpose of data representation, linear algebra can be used to retain significant variance by collapsing high-dimensional data into lower dimensions. Such dimensionality reduction is crucial in large-scale network data for gaining insights on patterns reflecting security vulnerabilities.

○ Optimization Techniques: Machine learning algorithms work on minimizing cost functions and optimization is something which needs to be checked on whether model accuracy would increase or decrease. Some of the optimization techniques (Gradient Descent, Newton's Method and L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno)) modify the model parameters for training data detection better security threats[3].

○ Detecting a vulnerability through probability & statistics: Vulnerability detection often contains uncertain and incomplete information. For example, one will then try to employ probabilistic models like Bayesian Networks and Markov Chains to model the probability of vulnerabilities given observed data. There is a need for statistical analysis methods to determine the significance of anomalies, including clues in network traffic that provide evidence on detecting whether an activity that produced an anomaly is indeed a benign or malicious.

○ Using Graph Theory: Model Network security can be well modelled as graph structures, in which the nodes corresponds to entities (e.g., computers or users) and edges represent communication relationships or data streams. In graph-based approaches to network measurement centrality measures, communities can be used to diagnose vulnerabilities and honeypots opportunities of the network topology[4,5].

- Math Model Numerical Methods: (Used for Data Exploration) Differential equations and System of equations are used to model how network systems behave over time. Mathematical models help to replicate multiple conditions in the simulated habitation of security weaknesses over time.

● Machine Learning Models for Identifying Detecting Vulnerabilities:

By leveraging machine learning in conjunction with sophisticated mathematical methods, automated and smart vulnerability identification capabilities are realised. Model: Based on the problem at hand and the type of data, a suitable model can be chosen for security. In this article, we will discuss a few Antivirus machine learning models that when coupled with mathematics can prove beneficial as well and have the ability to increase cybersecurity.

- Income: Using Bayesian Networks to model the uncertainty in your data

Bayesian networks (BNs) are graphs used for expressing the probabilistic relationships between a set of variables. In information security, BNs can model the aspects of interdependency between different security features where vulnerability in one feature can depend on another vulnerabilities[6]. Among others, a Bayesian Network can be used to represent which network traffic patterns may suggest that for example certain safety hazards are developing. The poster mentions that by refreshing the network with new evidence (ex: looked-for knowledge of the running state observation data, gathered through network monitoring), Bayesian inference provides an ability to reassign the probability of suspected security events and helps in identifying emerging vulnerabilities. The integration of BNs requires application of advanced probability theory and statistics. For example, the Expectation-Maximization (EM) algorithm is a common way to estimate the model parameters in BNs when data is incomplete as well. It allows us to really leverage the power of probabilistic models in a way that lets us quantify our uncertainties and make decisions about potential security incidents.

- Decision Trees & Ensembles

This algorithm is very intuitive since it classify the data by recursively partitioning the input space with respect to feature values[7]. These can be useful when the dataset consists of some categorical and other numeric features. DTs can be improved by detection system.

Random Forests: Random forests is an ensemble method that trains several decision tree models on different subsets of the data and combines their predictions. Random Forests are less likely to over-fit due to the use of randomness in feature selection and data sampling, making them better at finding intricate security patterns.

Gradient Boosting (XGBoost for instance): This model sequentially trains shallow trees as weak learners and combine them to have a powerful model at the end. This makes it an effective tool for real-time bug identification and the use of techniques like regularization, available in XGBoost allows improving generalization[8].

The mathematical basis for these models is a mix of tree construction based on information theory (eg entropy and information gain) and optimization to incrementally tune the model accuracy.

Auto-encoders (Anomaly Detection): Auto-encoders are a class of neural networks designed for unsupervised learning tasks such as anomaly detection. They are trained on a compressed representation of the input data (encoding) and an attempt is made to reconstruct the data from this encoding (decoding). This is used as a gauge to check the anomalies in memory, which indicate the difference between the original data and its reconstruction. Autoencoders are trained on ordinary network traffic in the context of vulnerability detection[9]. In other words, when any unusual or anomaly traffic patterns (for security breaches) are introduced in the model, it then experiences a spike in reconstruction error i.e., outliers which could act as potential threats and can be recognised.

Probabilistic Variational Auto-encoders (VAEs): VAEs are an extension of traditional auto-encoders which employ a form of probabilistic inference in the encoding process[10]. Specifically, sparsely gated AUTOENCODERS are the normal auto-encoders with a learned scaling factor for all of the activation units that introduces more stocjhas has ticity to their output activations and enables them provide not just 1 literal encoding for each input but also learn a distribution over possible encodings, resulting in greater robustness for detecting anomalies in scenarios where variations of data is inherently stochastic.

Gaussian Mixture Models (GMM) for Unsupervised Clustering: GMMs are the generatively with unknown parameters. For example, in cybersecurity, GMMs can be used to cluster network traffic behaviors into categories like normal and suspicious to find activities that reveal security breaches[11].

The EM algorithm: To estimate the parameters of the Gaussian distributions, allowing the model to iteratively adjust these estimates while refining its clustering through successive iterations. This unsupervised method is especially helpful for detecting zero-day opportunities where labelled data often does not exist.

Isolation Forests for Anomaly Detection: Isolation Forests (IFs) are unsupervised anomaly detection method which isolates instances in a dataset. The main insight is that anomalies are points that are more easily modulated than normal points. In this model, the random decisions are decision trees and how many turns have to be made till we isolate a data point is its anomaly score. The insertion of randomness in the partitioning process permits approximating complex relationships between data points to be linearized and subsequently captured by simple tree based data structure such as binary tree (BST).When a sample is isolated, it implies that at least 1 feature value of this record is surprisingly off compared to other records. This allows them to be computationally fast, making them suitable for high-speed security data, in which the quick identification of anomalous patterns are essential[12].

Reinforcement Learning for Dynamic Security Monitoring: In cybersecurity, RL might find application in, e.g., dynamic vulnerability management keeps changing the strategy along with network adaptation and all new threats.

Q-Learning: It is a simple and powerful RL technique, with Q-learning the agent learns the best action to take due to state-action combinations and rewards are recorded for actions taken at each state using which an agent try to maximize reward over time[13]. In the realm of network security, an RL agent can learn to react to security-related events (e.g., intrusion attempts), updating its policy on-the-fly by incorporating observed outcomes and therefore improving real-time defences.

A mathematically sound framework: Despite the underlying complexities of security, Reinforcement Learning models most of its tasks using Markov Decision Processes (MDPs) and dynamic programming for policy optimization; thus enabling it to effectively be applied as a part of complex controlled decision making algorithms for security dependent domains.

- Combining Mathematical Methods with Machine Learning

Mathematical techniques are often an important component of a powerful vulnerability detection system, but they add clutter to the clean machine learning pipeline. Counter: Statistically analysing the input data to enhance its quality before it is fed into machine language models (better known as feature engineering). For example, one can represent network traffic as a graph and learn anomaly detection algorithms (e.g., Autoencoders, GMMs) on graph-structured data to detect vulnerabilities. Using optimization methods to tune the hyper-parameters of a machine learning model as to prevent overfitting and allow them to have good performance on new, unseen data. In addition, probabilistic models such as Bayesian Networks can be coordinated with other algorithms (e.g., merging the output of an autoencoder within a Bayesian setting) in order to generate hybrid models which improve the capability for accurate detection. Although the integration of high-end mathematical models and machine learning is a potentially exciting method for detection of information security vulnerabilities brings in its own set of challenges:

- Data Quality and Labelling: A lot of machine learning models actually depend on high-quality, labelled data sets, which are difficult to obtain in cyber security due to the fact that threats evolve quite rapidly. Future work will focus on exploring semi-supervised and unsupervised learning methods to tackle this problem.

- Scalability: Cybersecurity solutions are dealing with vast amounts of real-time data. Machine learning models must be computationally efficient to Scale them up to caches storing hundreds of millions or billions of pages on the one hand and ensure high detection accuracy at the same time.

- Explain-ability: Since deep neural networks are able to model human-like cognitive functions, it is key that decisions made by such highly complex models can be explained in the context of Security Operations. More investigations in explainable AI are required for accounting and justifying these vulnerability detection results.

- Continuous Learning: As attackers change their ways, models must always be up to date and adapt with the threat and network conditions online. In this regard, work on reinforcement learning and continual learning is very promising.

The combination of modern mathematical methods and machine learning provides a very strong weapon for detecting vulnerabilities in information security. In this paper, a novel way to detect vulnerabilities in highly dynamic complex systems dubbed Bayesian Networks, Autoencoders, Gaussian Mixture Models and Reinforcement Learning along with statistical/probabilistic methods was proposed. The adoption of these methods opens the door to advancing adaptive-security systems with better ability to cover both the previously-seen-to-known and known-to-unseen threats. This work can potentially evolve the cybersecurity frameworks in future to ensure privacy and security of critical information at a time when everything is connected.

## 2. Related work:

A combined effect of machine learning and sophisticated statistical methods is one of the strategies being successfully used for information security vulnerability detection. This section provides a compilation of the similar works on vulnerability detection with known mathematical and machine learning models. The literatures can be categorized into supervised learning, unsupervised learning, probabilistic model, anomaly detection and hybrid techniques based on the specific mathematical method used to strengthen security.

● Supervised Learning for Vulnerability Detection:

The effectiveness of supervised learning models is well studied in the context of vulnerability detection, especially when labeled datasets are available. Majority of the research have applied different algorithms namely Decision Trees, Random Forests, Naive Bayes and Neural Networks to classify security threats. These models are trained on labelled data so they are able to spot distressed vulnerabilities by learning their features. Feature based vulnerability detection Random Forests over network traffic (with attribute entropy and statistical moments)[14]. These extracted features were used to update a classifier which was then able to well identify known vulnerabilities. While these models perform quite well they come with limitations, particularly when it comes to zero-day vulnerabilities that have less amount of labeled data for new threats. The efficacy of Artificial Neural Networks (ANNs) Deep Learning has been widely used in vulnerability detection with Fully Connected Neural Networks (FCNNs) and Recurrent Neural Networks (RNNs) proposed in recent years. A deep learning based approach with ANN to forecast the vulnerabilities using past security data. With the help of mathematical optimization algorithms like Stochastic Gradient Descent (SGD), they optimized a big neural network design working on high-dimensional data. Nevertheless, model performance was closely tied to the quality and size of sources of labelled training data but this only emphasized one problem that is the limited availability of good ground truth materials for supervised learning in cyber security[15].

● Unsupervised Learning and Clustering Methods:

With the paucity of labelled data in cyber security, unsupervised learning has gained significant importance in vulnerability detection. Such data does not have predefined labels and so these models are useful where the edge of detection is to find novelty or zero-day vulnerability.

○ Clustering based anomaly detection:

There have been a lot of work carried out to explore clustering algorithms like K-means, Hierarchical Clustering and Gaussian Mixture Models (GMMs) for detecting anomalies which can be equivalent with security threats. Using K-Means the network traffic data is clustered into different groups, assuming that normal case -doses region- makes dense clusters while anomalous cases (vulnerabilities) lies on sparse regions[16].

The clustering process can be mathematically described as follows. Given a dataset $X = \{x_1, x_2, \dots, x_n\}$ where each $x_i$ is a d-dimensional data point, K-Means seeks to minimize the total variance within clusters:

$$J = \sum_{j=1}^{K} \sum_{x_i \in C_j} \parallel x_i - \mu_j \parallel^2,$$

Where, $K$ is the number of clusters, $C_j$ is the set of points in cluster $j$, and $\mu_j$ is the mean of points in $C_j$. This objective function $J$ represents the sum of squared distances between each data point and its cluster centroid, effectively separating normal data from outliers.

To model the network traffic as a Gaussian Mixture Distribution (GMD), we focus on using Gaussian Mixture Models (GMMs). GMMs, through the Expectation-Maximization (EM) algorithm, are capable of iteratively finding more accurate parameter estimates for the underlying Gaussian distributions representing regions of traffic in a network that may showcase unusual patterns hinting at vulnerabilities.

○ Anomaly Detection using Autoencoders

An autoencoder is a type of neural network used in unsupervised learning with high success, as far as cybersecurity goes. They learn to reconstruct the input data and then measure the reconstruction error to pick up on anomalies. To detect vulnerabilities, researchers used Autoencoders with reconstructed network traffic data so that the normal traffic syndicated a low reconstruction error and the anomalous one (thus potential vulnerabilities) a high reconstruction errors[17].

Autoencoders will use math magic to squeeze the data into this lower-dimensional latent space and then re-expand it back out to its original number of dimensions. In mathematical term,

$$L(x, \hat{x}) = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2,$$

Where, $x$ is the input data, $\hat{x}$ is the reconstructed data, and $n$ is the number of features. The goal is to minimize this loss function during training, leading to the ability to identify anomalies when reconstruction errors deviate significantly from the norm.

● Vulnerability Detection with Probabilistic Modelling

One popular class of models in the vulnerability detection is probabilistic models such as Bayesian Networks, Hidden Markov Models (HMMs), which are particularly well-suited to modelling uncertainty in security data[18].

○ Dependency Modelling using Bayesian Networks

Example: Bayesian Networks to Represent Interdependencies of Security Features Infrastructures for cybersecurity6 can help model the interdependency between different security features and their impact on vulnerabilities in e.g., network traffic patterns, user activities etc. a model of Bayesian Network for predicting Software’s vulnerable-ness supported on historical vulnerability data and software metrics.

A Bayesian Network is defined mathematically using the conditional probability distributions for every node in the network for set $X = \{X_1, X_2, \dots, X_n\}$,

$$P(X) = \prod_{i=1}^{n} P\big(X_i \mid Parents(X_i)\big),$$

- Modelling for Temporal Analysis using Hidden Markov Models

Hidden Markov Models (HMMs) have been used to get a model for describing time-evolving vulnerabilities so that they can adjust in real-time with behaviours of hidden states. Modelled network states and detected transitions that are symptoms of security breach using HMMs[19]. Using the Baum-Welch algorithm, they were also able to find parameters of HMMs for sequences of events corresponding to possible exploitation. HMMs is a mathematical framework that uses hidden states to represent dynamic systems, where the observations are assumed to be probabilistic given those hidden states. The model's parameters can be learned from historical data (to capture relationships among states-transition probabilities and emission probabilities) enabling the vulnerability detection mechanism in real-time.

- Hybrid Models and Ensemble Techniques:

Studies have shown that using ensemble methods combining multiple machine learning models can increase the reliability of detection systems with respect to vulnerabilities. Here, Ensemble learning techniques like Boosting, Bagging & Stacking are used to generate predictions from multiple models for improved accuracy and lesser false positives.

- Boosting algorithms – XGBoost

XGBoost is an implementation of gradient-boosted decision trees which has been successfully used in cybersecurity due to the high performance and precise detection[20]. Vulnerability identification through classification network traffic and application of XGboost iteratively constructs trees to correct the errors of prior trees, making it suitable to learn from diverse patterns present in security data.

It is the same reason why XGBoost's objective function can be written as:

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

Where, $l(y_i, \hat{y}_i)$ is a loss function (e.g., squared error) measuring the difference between the predicted $\hat{y}_i$ and the actual $y_i$, and $\Omega(f_k)$ is a regularization term that controls the complexity of the model. This combination of mathematical optimization and model regularization enables XGBoost to achieve high performance in vulnerability detection tasks[21-25].

- Probabilistic and Deep Learning Hybrid Models

Hybrid approaches that combine probabilistic models with deep learning have been proposed to overcome these limitations of individual model types. The model itself combines the best parts of a VAE with Bayesian Networks. Inference: Both the VAE compresses input data into a latent space which captures salient features, and the Bayesian Network captures possible probabilistic dependencies among these features to predict vulnerabilities. Performing these operations in combination utilizes the forces of deep learning in feature extraction, together with probabilistic models for uncertainty management, boosting vulnerability detection performance accordingly.

- Dynamic Vulnerability Management with Reinforcement Learning

In IoT, cybersecurity, a specific combination of supervised and reinforcement learning (RL) approaches that allow real-time event based vulnerable member detection and adaptable structure has been employed. For this reason, in recent years, RL models (and Q-Learning in particular) have been used to create agents that interact with network environments and learn which policies are better at opposing vulnerabilities. Used Q-Learning to learn from observed network state methods and thus allowing the defence strategy of their reinforcement learning agent to adapt by providing optimized actions to avoid security breaches with each action taken.

The heart of Q-Learning lies in the way it computes and updates its Q-values which is modelled after the Bellman equation,

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)],$$

Where, $Q(s,a)$ is the value of taking action $a$ in state $s$, $\alpha$ is the learning rate, $r$ is the reward, $\gamma$ is the discount factor, and $s'$ is the next state. This formulation enables the agent to learn optimal actions over time, dynamically enhancing network security[26,27].

Reviewing the related work also shows that there are many machine learning models and mathematical methods applied information security vulnerability detection. Supervised methods -Random Forests and Neural Networks- in case that we have labeled data, unsupervised as K-Means and Autoencoders for new attacks and zero-day vulnerabilities. Bayesian Networks are very well suited for modelling uncertainty, and HMMs can be represented as a special case of Bayesian Networks (defined by a CPD representation). Furthermore, ensemble methods and hybrid models combine the power of several algorithms to improve the detection accuracy. Though, one can also face challenges like less availability of labelled data, high dimensionality of network traffic, and the requirement to have adaptable models aware to recent security threats. Combining advanced mathematical techniques with machine learning can open new horizon into how we can solve these challenges and in particular towards more intelligent and adaptive cybersecurity systems. The future directions for work on this issue probably centre around the development of more efficient hybrid models, better interpreting complex algorithms, and advancing real-time adaptive vulnerability detection systems.

## 3. METHODOLOGY:

- Data Collection & Pre-processing:

Sources of Data: The formulation asks for a variety of top-notch datasets to create a vulnerability detection system. Common data types used in cybersecurity include network traffic data, system logs and firewall records, and endpoint security data. In fact, publicly available datasets like the CICIDS 2017 dataset (composed of a diversity of network traffic data) and the UNSW-NB15 dataset (comprised of labelled instances by normal and malware activity) for this research can also be referenced. You can also collect live data from a network of the organisation using network monitoring tools such as Wireshark or Zeek to build customized datasets. This may allow us to apply supervised, unsupervised and semi-supervised machine learning methods, using labelled and unlabelled data.

- Data Pre-processing:

Network data can often be noisy, include missing values and irrelevant variables. Hence, it is an important stage in the entire process of natural language processing which consisted of the following sub-processes:

Missing values in the data cleaning, Can remove or impute them. Median or mean imputation can be used for replacing missing values in case of numerical attributes. Another way for an imputation for the missing values for categorical variables that can be mode.

○ Outlier Removal: Detect and delete large outliers that can be considered as noise or corrupted records according to statistical techniques. Outliers, for example, can be identified using dimensionality technique called the Z score method.

$$Z_i = \frac{x_i - \mu}{\sigma},$$

Where, $Z_i$ is the Z-score of data point $x_i$, $\mu$ is the mean of the dataset, and $\sigma$ is the standard deviation. Entries with $| Z_i | > 3$ are considered outliers.

○ Impute missing values: Use mean/mode substitution, K-Nearest Neighbours (KNN) etc. to handle imputation of missing values in the dataset.

Feature Engineering: Packet size, duration, protocol type, source / destination IP address and port numbers. This type of information entropy is an important domain-specific feature, as it can be used to differentiate between network traffic acts (i.e., entropy of traffic) also, second-order statistics (e.g., mean and variation), and some flow-based metrics such as the amount of packets transferred over individual flows or the number of requests.

○ Principal Component Analysis (PCA): PCA reduces dimensionality by transforming the original data into the new orthonormal set of features. We compute the covariance matrix:

$$\Sigma = \frac{1}{n-1}(X - \mu_X)^\top (X - \mu_X),$$

Where, $\mu_X$ is the mean vector of $X$. Eigenvalues and eigenvectors of $\Sigma$ are then used to form a reduced feature space, retaining components with the highest variance.

○ Mutual Information: Calculates mutual information value between each categorical feature and target variable to choose top performing features. Where the mutual information is:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right).$$

Normalization: Normalization will be used some normalization method like Min-Max scaling or Z-score normalization to normalize the attribute. This method is useful when we perform this type of algorithm which is very sensitive to the scale of data like the algorithms scaling with a fixed length output.

Since the number of features for network traffic data are very high, PCA has been used to reduce the feature space while retaining maximum variance in the data. In mathematical terms, PCA projects the input data into a lower-dimensional subspace by finding eigenvectors of the covariance matrix of the input data. This helps alleviate the “curse of dimensionality,” and will help you train your machine learning model better.

- Model Selection and Mathematical Machines

This section introduces several machine learning models, in each of which a set of different mathematical operations are applied to improve the security detection capability. The models, in turn, are chosen to protect against various kinds of cyber security threats from known vulnerabilities to zero-day exploits.

Supervised Learning Models:

For this purpose, the supervised learning approach is quite helpful if there are enough labeled datasets in line with the malicious traffic involved. These models extract patterns hidden in the labelled data, and use them to classify new instances that were not seen during the training of the model.

- Logistic Regression (LR):

Logistic Regression: Logistic Regression is a linear model used for binary classification problems. The logistic regression models the probability that a binary outcome is 1.

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d)}},$$

Where, $\beta_0, \beta_1, \dots, \beta_d$ are the model parameters learned during training, and $X_1, \dots, X_d$ are the input features. The model is trained by maximizing the likelihood of the observed outcomes, which is mathematically equivalent to minimizing the cross-entropy loss function:

$$L(\beta) = -\sum_{i=1}^{n} [y_i log P(y_i) + (1 - y_i) log\big(1 - P(y_i)\big)].$$

Logistic Regression provides interpretable results, with the coefficients $\beta_j$ indicating the impact of each feature on the vulnerability prediction. However, it may struggle with complex relationships in high-dimensional data, necessitating more sophisticated models.

- Decision Trees and Ensemble Techniques

In decision tree data is divided among subgroups based on feature values using either Gini Impurity or Information Gain measures for selecting a split. The Gini Impurity for a binary split is represented by:

$$G = 1 - p_1^2 - p_2^2,$$

Where, $p_1$ and $p_2$ are the proportions of the two classes in a node. Decision Trees recursively partition the feature space, creating a tree structure that assigns labels to data points based on the learned rules.

Algorithm for random forest:

1. Input: Training dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.
2. For each tree in the forest:

Sample $N$ data points from $D$ with replacement.

Select a random subset of features.

Construct a decision tree based on the selected features.

3. During prediction, aggregate the results from all trees (majority voting).
4. Output: Predicted class for the input data.

Random Forests use information gain or Gini impurity to split nodes in each decision tree. For Gini impurity, the split criterion is:

$$G = 1 - \sum_{i=1}^{C} p_i^2,$$

where $p_i$ is the probability of selecting a class $i$ from the node, and $C$ is the total number of classes.

Ensemble Methods: Don't put all your eggs in one basket Ensemble methods, like Random Forests and Gradient Boosting (XGBoost), build multiple decision trees to increase performance of the model. Random Forests are a special case of bagging that create an ensemble of trees by randomly choosing on every tree both data (a bootstrapped sample of the training data) and features, reducing overfitting. The final prediction g (IB,) is simply the average (for regression) or majority count (for classification) of the predictions from these trees: A(B,,,, θm).

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(X),$$

Where, $T_i(X)$ is the prediction of the $i$-th tree, and $N$ is the total number of trees.

Decision trees in Gradient Boosting are built one after another and tree building process minimizes errors of its predecessor. In Gradient Boosting, The gradient boosted tree minimizes the objective function which is made up of a loss function LLL and a regularization term to control model complexity. If we are to say this as an example, object is something like the following in XGBoost.

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

Where, $l(y_i, \hat{y}_i)$ is the loss function (e.g., squared error for regression) and $\Omega(f_k)$ is a regularization term that penalizes the complexity of trees $f_k$.

o XGBoost: Extreme Gradient Boosting

XGBoost (Extreme Gradient Boosting) is a robust ensemble learning method that builds an ensemble of decision trees for predicting vulnerabilities in network traffic. This method uses gradient boosting, where the errors of the previous trees are corrected by each new tree uphill. The regularized objective function of the method is represented as:

$$L(\theta)=\sum_{i=1}^{n} l(y_i,\hat{y}_i)+\sum_{k=1}^{K} \Omega(f_k),$$

Where, $l(y_i,\hat{y}_i)$ is the loss function measuring the difference between the predicted output $\hat{y}_i$ and the actual label $y_i$, and $\Omega(f_k)$ is a regularization term that controls the complexity of the model.

XGBoost uses regularization to prevent overfitting, and its objective function is defined as:

$$L(\theta)=\sum_{i=1}^{n} l(y_i,\hat{y}_i)+\sum_{k=1}^{K} \Omega(f_k),$$

Where, $l$ is the loss function, and $\Omega(f_k)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$ is the regularization term for tree $k$.

○ Neural Networks (NN) and Deep Learning:

Deep learning approaches, in particular Fully Connected Neural Networks (FCNNs), are used for complex pattern recognition in high-dimensional data. A neural network architecture consists of an input layer, one or more hidden layers and an output layer. It performs backpropagation to optimize weights via the mathematical formulation:

$$W_{ij}\leftarrow W_{ij}-\eta\frac{\partial L}{\partial W_{ij}},$$

Where, $W_{ij}$ are the weights connecting neurons, $\eta$ is the learning rate, and $L$ is the loss function (e.g., cross-entropy for classification tasks). The Stochastic Gradient Descent (SGD) algorithm is commonly used to update the weights iteratively.

● Unsupervised Learning Models

These unsupervised models are important because while we are trying to develop a supervised model, it is likely that the labelled data for zero day vulnerabilities are going to be sparse or not even available. It learns the normal patterns in your data so that it can spot outliers or any deviations from these learnt norms which may be a security threat.

○ Dead simple auto-decoders for anomaly detection:

One application for creating anomaly detection models is the auto-encoder neural networks, which are specially designed for unsupervised learning. In this architecture, they encode the input data into a latent representation with lower dimensionality and thereafter decode it back to the original structure. The above reconstruction error can be taken as an anomaly indicator;

$$L(x,\hat{x})=\sum_{i=1}^{n} (x_i-\hat{x}_i)^2,$$

Where, $x$ is the input data, $\hat{x}$ is the reconstructed data, and $n$ is the number of features. A high reconstruction error signals an anomaly, potentially indicating a vulnerability.

Algorithm:

1. Input: Dataset $X = \{x_1, x_2, \dots, x_n\}$.
2. Randomly select a feature and a split value for each tree.
3. Recursively partition the data until each point is isolated.
4. The anomaly score for a point $x_i$ is computed based on the path length to isolate it.
5. Output: Anomaly scores for all data points.

- K-Means and Gaussian Mixture Model as Clustering Techniques:

K-Means Clustering: K Means has been around since 1967, this method partitions data into KKK clusters based on similarity of features that are available. Which reduce Within-Cluster Sum of Squares.

$$J = \sum_{j=1}^{K} \sum_{x_i \in C_j} \| x_i - \mu_j \|^2,$$

Where, $C_j$ is the set of points in cluster $j$ and $\mu_j$ is the cluster centroid. In vulnerability detection, normal traffic forms dense clusters, while outliers (anomalies) represent potential threats.

Next example implements a Gaussian Mixture Models (GMMs), which is a probabilistic model that assumes all clustered data points are generated from several Gaussian distributions. The Expectation-Maximization (EM) algorithm is employ method to infer the parameters of these distributions. GMMs can characterize complex data structures, and thus they are appropriate to capture subtle anomalies in network traffic.

- Probabilistic Models

Developing probabilistic models is beneficial to its capability of handling uncertainty and dependencies in the different characteristics of network traffic data making it more reliable in detecting complex vulnerabilities.

- Bayesian Networks

Bayesian networks are a type of graphical model for representing probabilistic relationships among a set of variables. The mechanism is Bayesian inference, combining priors and evidence to get the probability of security events using observations. Here is the joint probability distribution of our network:

$$P(X) = \prod_{i=1}^{n} P\big(X_i \mid Parents(X_i)\big),$$

Where, $X = \{X_1, X_2, \dots, X_n\}$ is the set of variables, and $P\big(X_i \mid Parents(X_i)\big)$ is the conditional probability of $X_i$ given its parent nodes in the network.

Algorithm:

Input: Set of variables $V = \{X_1, X_2, \dots, X_n\}$ representing security features.

Define the network structure as a Directed Acyclic Graph (DAG).

Compute the conditional probability tables (CPTs) for each node $X_i$: If $X_i$ has parents $Pa(X_i)$, calculate $P\big(X_i \mid Pa(X_i)\big)$.

During prediction, use Bayes' theorem to update the posterior probabilities:

$$P(X_i \mid E) = \frac{P(E \mid X_i)P(X_i)}{P(E)},$$

Where, $E$ is the observed evidence.

Output: Updated probabilities indicating the likelihood of vulnerabilities.

- Graph Neural Networks (GNNs)

Graph neural networks have been designed toward graph-structured data which enables us to perform network traffic analysis as well as vulnerability detection in complicated systems. One node in the GNN uses the message-passing mechanism to aggregate information from its neighbours on the feature vector.

Mathematically, the feature vector $h_v^{(l+1)}$ of node $v$ in layer $l+1$ is updated as:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \sum_{u \in N(v)} h_u^{(l)}\right),$$

Where, $N(v)$ is the set of neighbors of $v$, $W^{(l)}$ is a learnable weight matrix, and $\sigma$ is an activation function. This process enables the model to learn node representations that capture local and global graph structures, aiding in the identification of vulnerabilities across network nodes.

- Hidden Markov Models (HMMs):

In particular, hidden Markov models (HMMs) have been applied to represent the temporal dependencies amongst network traffic, as part of an effort to describe how the global state of a network evolves over time. The two comprise of hidden states and observable emissions, which are known as state transition probabilities and emission probabilities. To achieve this, we use the Baum-Welch algorithm to estimate the parameters of the HMM.

The model can be described as follows:

$$P(X, S) = P(S_1) \prod_{t=2}^{T} P(S_t \mid S_{t-1}) \prod_{t=1}^{T} P(X_t \mid S_t),$$

Where, $S_t$ represents the hidden state at time $t$, and $X_t$ is the observed data at time $t$.

- Hybrid Models

This often increases the potential powers of the vulnerability identification as one may say by combining multiple models. As an example, a hybrid system might combine an Auto-encoder for anomaly detection with the XGBoost of type of vulnerabilities identified.

- Optimization and Training in Machine Learning

Model training is the process of tweaking the model parameters in order to reduce a selected loss function. And it is because we need this generalization in our model to be able to detect problems that not only the ones seen during training.

○ Optimization Techniques:

Gradient Descent: An iterative minimization algorithm that is used to find the set of parameters which gives the minimal value of loss function.

$$\theta \leftarrow \theta - \eta \nabla L(\theta),$$

Regularization: Regularisation is an important weapon to prevent overfitting, two most common are L1 (Lasso) and L2 Ridge regularisation. In L2 regularization, term which is controlled by C is the sum of square value of weights and bias while optimizing the loss function waged with this parameter.

$$L_{reg}(\theta) = L(\theta) + \lambda \sum_{j=1}^{d} \theta_j^2,$$

Where, $\lambda$ is the regularization parameter controlling the strength of the penalty.

Implementation can be done in Python and we can use libraries such as scikit-learn (For XGBoost, clustering and basic models), TensorFlow or PyTorch( For deep learning models like Auto-encoders, FCNNs) for developing our model, pandas for data pre-processing. Specialized libraries such as pgmpy (for Bayesian Networks) and hmmlearn (for Hidden Markov Models) that can be used to implement probabilistic models. Hyper-parameter tuning helps enhance the model performance. Several ways are taken to find all the best hyper-parameters for these models like Grid Search, Random Search etc. Early stopping and learning rate decaying are inevitable techniques for a well-behaved training of deep learning model.

The purpose of this methodology is to introduce a unified approach to information security vulnerability detection, by leveraging advanced mathematical techniques and machine learning models. This methodology also mixes supervised learning (e.g., XGBoost), unsupervised learning (e.g., Auto-encoders, clustering) and probabilistic models (e.g., Bayesian Networks, HMMs) to develop a versatile system able to identify a broad spectrum of vulnerabilities including zero-day attacks. It also enhances the immunity of detection process by including hybrid models and mathematical methodologies aver using PCA for dimensionality reduction and Bayesian inference. This complete framework can be tailored and implemented for real-world security use cases, allowing organizations a smart adaptive security solution.

## 4. RESULTS:

The study presents an image of the same machine learning models and approaches for vulnerability detection, as practiced in data from different terms good points and bad ones. They have been tested on popular datasets, like CICIDS 2017, UNSW-NB15, custom live network data as well as old ones (KDD Cup 99 and DARPA 2000) trying to demonstrate the adaptability to both recently detected and future security challenges. The results indicate that machine learning combined with advanced mathematical techniques can be powerful for the real-time identification of vulnerabilities, including zero-day threats in network environment.

- Supervised Learning Models

XGBoost also dominates over all other models in accuracy, precision and recall on all datasets in the domain of supervised learning. Largest accuracy has been obtained for custom live data- set (97.1%) and CICIDS 2017 (96.8%), representing its strength in utilizing network environments of different types. The strength of XGBoost is that it can process high-dimensional data since keeping in view feature-reduced data method PCA on completion then used. The model also excelled in providing low false positive rates (1.5–2% across varied datasets) over time, which suggests that it will be capable of maintaining this same level of trustworthiness if deployed at scale to monitor networks in real-time.

Fully Connected Neural Networks which have shown robust detection performance on even the challenging cases of network data in FCNNs and traditional back-doored neural networks were mostly present otherwise. Accuracy: although both reliable and stable, there was no real breakthrough in accuracy improvement over the already excellent 94.8% achieved by XGBoost; a false positive rate varied somewhere between 2.3–3.2%, but couldn´t beat out XGBoost here, either; This is a strong evidence that deep learning models outperform traditional ML algorithms for capturing non-linear relationships in the data, whereas ensemble methods like XGBoost lie approximately in between deep and linear learner when it comes to accuracy-interpretability trade off.

| **Model** | **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **Training Time (s)** | **False Positive Rate (%)** |
|---|---|---|---|---|---|---|---|
| XGBoost | CICIDS 2017 | 96.8% | 96.1% | 97.2% | 96.6% | 150 | 1.8% |
| | UNSW-NB15 | 95.5% | 94.7% | 96.0% | 95.3% | 200 | 2.0% |
| | Custom Live Dataset | 97.1% | 96.5% | 97.8% | 97.1% | 220 | 1.5% |
| Neural Network (FCNN) | CICIDS 2017 | 93.2% | 92.4% | 94.0% | 93.2% | 180 | 2.5% |
| | UNSW-NB15 | 92.0% | 91.1% | 93.5% | 92.3% | 230 | 3.2% |
| | Custom Live Dataset | 94.8% | 94.0% | 95.3% | 94.6% | 250 | 2.3% |

The following table summarizes the model performance and computational requirements of XGBoost and Neural Networks using various datasets, such as Accuracy, Precision, Recall, F1-Score, Training Time In Sec(E), False-Positive Rate, etc.

- Unsupervised Learning Task: Anomoly Detection

Flexible custom unsupervised models performed with greater variance in anomaly and zero-day detection. The autoencoders, and a probabilistic variant called Variational Auto-encoders (VAEs), achieved high precision-recall for all detectors, especially the custom live dataset (up to 94% F1-score with VAEs). The models running with the reconstruction error-based approach were able to nicely separate benign and malicious patterns of traffic. Nevertheless, broader detection capabilities against more nuanced anomalies without ground truth labeling position this class of models as a key element in real-time monitoring systems, especially for highly dynamic environments with rapidly changing attack patterns (albeit at the price of slower response times: IOCs typically take 10–20 ms per instance to predict).

| **Model** | **Dataset** | **Precision** | **Recall** | **F1-Score** | **Detection Time (ms/instance)** | **False Positive Rate (%)** |
|---|---|---|---|---|---|---|
| Auto-encoder | CICIDS 2017 | 90.5% | 92.8% | 91.6% | 12 | 3.1% |
|  | UNSW-NB15 | 89.8% | 91.5% | 90.6% | 15 | 3.5% |
|  | Custom Live Dataset | 92.2% | 94.0% | 93.1% | 18 | 2.8% |
| Variational Auto-encoder | CICIDS 2017 | 91.7% | 93.3% | 92.5% | 14 | 2.9% |
|  | UNSW-NB15 | 90.8% | 92.2% | 91.5% | 16 | 3.2% |
|  | Custom Live Dataset | 93.0% | 95.0% | 94.0% | 20 | 2.5% |
| Isolation Forest | CICIDS 2017 | 88.4% | 90.0% | 89.2% | 10 | 4.0% |
|  | UNSW-NB15 | 87.2% | 88.9% | 88.0% | 12 | 4.3% |
|  | Custom Live Dataset | 89.1% | 90.7% | 89.9% | 15 | 3.7% |

Here we have the performance of unsupervised models (Auto-encoder, Variational Auto-encoder and Isolation Forest) on different datasets with their precision, recall, F1-score, detection time and False Examples

Isolation Forests and Gaussian Mixture Models (GMM) achieved similar sensitivity but with a slightly lower accuracy level compared to Auto-encoders. GMMs, measuring network traffic as the superposition of Gaussian distribution functions, were evaluated with an F1-score ranging between 85 and 90% on different datasets, revealing their effectiveness for modelling network behaviour that could signal security threats. Yet their false positive rates (3.7 and 4.3%) are high which could be improved by further tuning for deployment in changing networked settings.

- Models of Probabilistic and Reinforcement Learning

The model based method employed Bayesian Networks which allowed a more flexible approach to vulnerability detection in the sense that was able to capture how different network features are relate probabilistically. Under this framework, the inference accuracy was 88.5%~91% over all samples in grapefruit-dataset, orange-dataset and lemon-dataset, which demonstrates the capacity of dealing with uncertain and incomplete information of each sample within network traffic data. Convergence to the reassessment rates of 94–97% for network configurations that change quickly (dynamic behaviour), which is a very useful property in face of rapidly changing security threats, were emphasised. Yet the training times were fairly long (200-300 seconds) to show that there was a trade-off between computational complexity and real-time ability.

| Dataset | Variables (Nodes) | Avg. Conditional Dependencies | Training Time (s) | Inference Accuracy | Dynamic Reassessment (%) |
|---|---|---|---|---|---|
| CICIDS 2017 | 25 | 4 | 200 | 89.2% | 96% |
| UNSW-NB15 | 30 | 3 | 250 | 88.7% | 94% |
| Custom Live Dataset | 35 | 5 | 300 | 91.0% | 97% |
| KDD Cup 99 | 28 | 4 | 220 | 88.5% | 95% |

Table gives the information related with the Bayesian Network model of different datasets like no of variable used, average conditional dependencies, training time, inference accuracy and dynamic probability reassessment capability.

More importantly, reinforcement learning models, namely Q-Learning was found to be efficient in providing dynamic security monitoring with a detection accuracy upto 93% on average. High rates of real-time adaptation (92–95%) demonstrate the inherent capability of the agent to learn optimal actions under differing network states and its ability to update its policy according to action outcomes are verified. On the convergence times (300 – 400 episodes), this seems reasonable for real-time defence mechanisms provided one has enough initial computational resources to bootstrap.

- Hybrid Models

Finally, Hybrid which merged Auto-encoder and XGBoost into a single model showed the best performance for all cases with an overall accuracy rate of at most 98.2 on custom live dataset while keeping a low false-positive rate (1.2–1.8%). While the model minimize false alarm and detected all of these, which tried to stay completely undetected it is worth to mention that 53% of which still were flagged by Autoencoder. The combined resolution of this hybrid method significantly speeds up the detection and brings the average inference time to around 10-15 milliseconds per instance, which is ideal for real-time applications.

we demonstrate the performance of the hybrid model on different datasets and report several metrics including accuracy, precision, recall, F1-score, false positive rate (FPR), average training time (over three runs) as well as detection time per instance.

| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **False Positive Rate (%)** | **Training Time (s)** | **Detection Time (ms/instance)** |
|---|---|---|---|---|---|---|---|
| CICIDS 2017 | 97.5% | 97.0% | 98.0% | 97.5% | 1.5% | 250 | 12 |
| UNSW-NB15 | 96.8% | 96.2% | 97.4% | 96.8% | 1.8% | 280 | 15 |
| Custom Live Dataset | 98.2% | 97.7% | 98. | | | | |

- Scalability and Computational Efficiency.

These models have also been subjected to scalability tests over a wider range of dataset sizes (from 500,000 to 5,000,000 records), showing their capacity of handling large-scale network data effectively. As the testbed datasets were grown from 500 K records to 5M records, we observed performance times for XGBoost (which showed reasonable training times, scaling from ~70 seconds for 500 K and ~800 seconds for 5 M) while Auto-encoders' performance time increased more modestly in relation to dataset size keeping them practical real-time detection. PCA-based dimensionality reduction helps in the faster training of models — 15–120 seconds depending on different dataset sizes which proves that model accuracy can be optimized without affecting its performance.

| **Dataset Size (Records)** | **XGBoost Training Time (s)** | **Autoencoder Detection Time (ms/instance)** | **Q-Learning Convergence (Episodes)** | **PCA Computation Time (s)** |
|---|---|---|---|---|
| 500,000 | 70 | 5 | 200 | 15 |
| 1,000,000 | 150 | 8 | 300 | 30 |
| 2,000,000 | 250 | 10 | 400 | 50 |
| 3,500,000 | 500 | 20 | 500 | 80 |
| 5,000,000 | 800 | 30 | 700 | 120 |

For different data set sizes, this table shows the computational efficiency and scalability of the models for training time, detection time, episodes to converge for Q-learning and PCA computation time.

The findings highlight that each model has its own strengths, depending on the type of cyber security scenario. XGBoost, as a supervised model, works really well with labelled data and we get accurate results out of it as well false positives are low. Without labelled data, unsupervised models especially auto-encoders make it very useful for identifying anomalies and zero-day threats. In a connected system, models like Bayesian Networks and reinforcement learning algorithms help maintain robust, adaptive security frameworks with the ability to grow along with changes within the network. Thus, the hybrid model (anomaly detection aided by advanced classification techniques) with appropriate improvements has displayed the best performance overall and appears to be a promising solution for end-to-end real-time evaluation of vulnerability. One important implication of these results is that combining a wide array for machine learning techniques and mathematical foundations can improve the robustness and efficiency of cybersecurity defence layers.

## 5. CONCLUSION:

The complexity and volume of cyber threats is increasing, therefore traditional methods to detect information security vulnerabilities as part of vulnerability management are not enough. This method allows to automatically identify and prevent known threats using signature-based recognition but also it enables devices with intelligent security capable of identifying zero-day vulnerabilities. The contribution made by this research was to investigate in what way the latest machine learning models can be combined with advanced mathematical techniques to improve upon the effectiveness of security vulnerability detection of information systems. Via this journey, we brought together models rich and diverse: from supervised learning to unsupervised learning; as well as probabilistic and hybrid modals supported by mathematical tools such optimization, probability inference, dimensionality reduction etc. These results indicate the importance of mathematics and machine learning in constructing resilient, intelligent systems that can detect vulnerabilities as they happen.

- Improving Detection with Advanced Mathematical Techniques

Modern machine learning algorithms use so much math on so many points of the vulnerability detection. Level up the Machine Learning modelsData preprocessing to Model optimization, Mathematical methods guide introduce a new beginning for overall performance of machine learning model. Namely, linear algebra techniques like Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)—used for dimensionality reduction of network traffic data reduced the "curse of dimensionality" yet retained pivotal variance for model learning. We found that this reduction not only sped up the training process but also increased accuracy and generalization, illustrating how mathematical approaches can be used to tame massive cybersecurity datasets with high dimension and scale. During the training of other models, these cost functions and optimization methods were minimized using Gradient Descent (and its variants including Stochastic Gradient Descent), Lasso Regression, etc., to fit model parameters. These strategies allowed us to ensure that machine learning models converged successfully and were able to detect complex security threats at maximum potential --- Moreover, probabilistic models such as Bayesian Networks used probability theory to solve the ambiguous and missing information to afford a state-based retelling of the net by contemplating new evidences. Markov Chains and Hidden Markov Models existed as a mathematical foundation for modelling time-series data that could be used to recognize vulnerabilities from temporal patterns in network traffic.

Also, graph theory improved network security modelling by casting (encompassing) Network entities (like computers and users etc.) as nodes, while interactions are branded as edges. This enabled us to use centrality measures and community detection algorithms to extract the vulnerabilities, attack paths i.e vulnerabilities exploited in series) that exist on the network. Note that these math-based techniques have become essential to building the machines learning models capable of spotting vulnerabilities in real-time.

- Watching-over for Existing Vulnerabilities Using Supervised Learning

Apply supervised learning (XGBoost, neural networks) very well to a labeled data set It was the use of an ensemble learning method based on decision trees and something else (gradient boosting) that allowed XGBoost to handle learn from both categorical as well as numerical data types, and potentially model more complex interactions between security features. It demonstrated high accuracy rates on diverse datasets in all network environments illustrating its consistency. Built-in feature importance analysis for the model could identify the most important attributes associated with security breaches and allowed more focused threat mitigation efforts. The issue is that the supervised learning models require some sort of labelled datasets which are very rare in cyber security area because it always changes. Strong classification ability, but closely related to the quality of the training data: For examples, Fully Connected Neural Network (FCNN) has a promising performance using its abundant learnable parameters but they need labeled in million scale. Even by their very nature, these are weak against zero-day threats and they highlight a key problem in cybersecurity: the requirement of supervised prediction techniques that depend on labeled data.

- Zero-Day Vulnerability Detection with Unsupervised Learning

In order to address zero-day or unknown vulnerabilities, we studied unsupervised models to include Auto-encoders, VAEs, Isolation Forest and GMM. Auto-encoders, which were trained to try and reconstruct the input data, performed well on this topic due to being able pick up deviations from normal network patterns. The spike in Reconstruction Error signalled potential areas of attack when new and malicious network traffic was introduced, so provided a way for bootstrapping to automatically detect new or unseen threats. Variational Auto-encoders which are the probabilistic extension of Auto-encoders bring stochastic input to the generation process and make it much more capable in handling diverse data. Both Isolation Forests and GMMs modelled the distribution of network traffic data, providing an alternative view towards anomaly detection. Isolation forests isolated anomalies by partitioning the data according to how much their features deviate, whereas GMM captured network activity distribution, and deviations from normal clusters were considered anomalous. Especially useful in cybersecurity environments with few labelled inputs, these unsupervised methods are not reliant on predefined categories by which to find vulnerabilities.

The wide range of success unsupervised learning models has reinforced the notion that effective security systems can be built to detect both known and emerging threats without needing an extensive labeled dataset. This agility is essential in ever changing threat environments where new vulnerabilities and attack vectors are discovered on a regular basis.

- Dynamic Security Event Monitoring with Probabilistic Models

Such a methodology enhances the performing accuracy of a method as compared to rule-based techniques, probabilistic models like Bayesian Networks and Hidden Markov Models (HMMs) brought uncertainty handling and adaptation in vulnerability detection which is complementing with our vagueness in natural language. Dynamic updates to the probabilistic relationships between security features were modelled using Bayesian Networks as new evidence became available. For example, the system is able to re-evaluate how likely certain vulnerabilities are based on what has been happening in terms of network traffic, allowing for an adaptive, real-time reaction according to current security situation. This probabilistic inference feature is a requirement for environments in which network behaviours may change without explicit warning and the forms of security threat are continuously shifting. HMMs expanded this strategy through modelling the evolution of network states over time and detecting temporal transitions that may hint at anomalies caused by a security breach. Because HMMs learned state transition and emission probabilities, we could pinpoint sequences of activities matching exploitation attempts. In that instance, the time-series analysis proved to be extremely powerful in identifying temporal weaknesses concerning the attacks placement over months of period with an eye on utilizing probabilistic models as a framework for building proactive security responses.

- Adaptive Vulnerability Management with Reinforcement Learning

Agents were developed with the help of reinforcement learning (RL), forest-calling Q-Learning which could execute dynamic vulnerability management. The RL agent interacted with the environment and maximized cumulative rewards over time to learn optimal actions for improved network security. The policy was adjusted to the outcomes and response strategies were honed over time on incorporating new threats. Q-Learning opened up the realm of real-time decision-making in complex, dynamic environments, and hence was a huge step in the direction of automated self-improving cyber security systems.

Reinforcement learning helps machine learning models act in addition to detecting against network security. We designed the RL agent to model security landscape through solving Markov Decision Processes (MDPs) and dynamic programming based policy optimization problems, so that it can sense different types of security threats and guide its intention towards mitigating the risk associated attacks, when it learns with simulated world. This is a huge benefit when dealing with security in the large-scale, distributed network environment where tight monitoring and quick response are necessary to keep solid security protocols intact.

- Improved Detection Accuracy with Hybrid Models

The combination of several machine learning models within a hybrid framework outperformed the rest, in that it achieved higher accuracy, precision and real time detection. For example, by using an Autoencoder for anomaly detection and then a semi-supervised XGBoost classifier that takes from the autoencoder the detected anomalies as input features—reconstruction errors (unsupervised), routes can be chosen which better leverage supervised learning in threat prediction. This hybrid approach achieved high performance accuracy, low false positive rates over several datasets (CICIDS 2017 standard benchmark, UNSW-NB15 standard benchmark and custom live network).

In reality, the success of hybrid models advocates for the necessity of a more layered approach to security. This system integrated multiple machine learning methods to provide a broad variety of threat detection, which included known vulnerabilities as well as zero-day exploits and everything in between. The mathematical methodologies of the PCA used for dimensionality reduction and Bayesian inference utilized by the hybrid approach were more powerful in terms of performance and interpretability. This combination provides a reference model of dynamic, intelligent cybersecurity frameworks to proactively and responsively detect threats in real-time.

- Limitations and Future Directions

Although the study showed that mathematical techniques can enhance machine learning in vulnerability detection, common issues remain. Data Quality and Labelling – This remains one of the most challenging areas. Most machine learning models require good quality, labelled data- which is often scarce in security as the landscape evolves and threats are frequently changing.There appears to be a lack-of exploration of more start-of-the-art second-stage/simple retrieval (on compact/native data representations) cleverness for semi/unsupervised learning directly on all probing architectures with little or low tagged/gold-standard data though.

This also creates a challenge for scalability, as the volume of data is generated in real-time in your cybersecurity systems. The models develop in this paper look quite promising for being able to scale up to large-scale datasets, but computational efficiency and resource management is critical for real world applicability. Our future work will try to extend these models and make them acceptable in cloud-based environments or distributed network architectures where it may scale well yet have appealing detection accuracy. Interpretability of complex models, especially deep neural networks, on the other hand continues to be an open problem. So understanding why the model makes decisions is a critical part of justifying actions and building better defences in security operations. Future-make pipeline efforts of explainable AI (XAI) and model interpretability integration are required to bridge the gulf between model accuracy and actionable insights.

Finally, the learnability of rapidly-evolving threats implied by their machine-based models; Methods such as reinforcement learning and continual learning show promise in developing systems that adapt with the threat landscape, but continued research is required to fully exploit their capabilities for cyber security contexts.

Concluding, the research presents an exhibition where mathematical models are applied with machine learning models to ensure vulnerability detection in information security. A unified approach to supervised learning, unsupervised learning, probabilistic modelling and reinforcement allows the complete nameless detection of known and unknown vulnerabilities in real time. Our results suggest that integrating multiple layers of mathematical techniques for data pre-processing, optimization and probabilistic inference can considerably improve both the performance and robustness of cybersecurity systems.

The results of this study seem to point the way for positive advancements in intelligent, adaptive cybersecurity frameworks. Whilst doing so, the cybersecurity community will hopefully progress on our path towards being able to develop systems which not only meet this requirement of identifying threats and vulnerabilities, but also have capabilities enabling us to make networks secure in an ever-changing digital landscape. These findings pave the way for a new breed of real-time alarm systems that will protect essential information by securing privacy and security in an age of growing dependence on digital connectivity.

**References:**

[1] Khan, Muskan, and Laiba Ghafoor. "Adversarial Machine Learning in the Context of Network Security: Challenges and Solutions." *Journal of Computational Intelligence and Robotics* 4.1 (2024): 51-63.

[2] Bharathi, V. "Vulnerability detection in cyber-physical system using machine learning." *Scalable Computing: Practice and Experience* 25.1 (2024): 577-591.

[3] Gong, Yulu, et al. "Enhancing Cybersecurity Resilience in Finance with Deep Learning for Advanced Threat Detection." *arXiv preprint arXiv:2402.09820* (2024).

[4] Yang, Peng, and Xiaofeng Wang. "Vulnerability extraction and prediction method based on improved information gain algorithm." *PloS one* 19.9 (2024): e0309809.

[5] Ozkan-Ozay, Merve, et al. "A Comprehensive Survey: Evaluating the Efficiency of Artificial Intelligence and Machine Learning Techniques on Cyber Security Solutions." *IEEE Access* (2024).

[6] Ni, Chunchun, and Shan Cang Li. "Machine learning enabled industrial iot security: Challenges, trends and solutions." *Journal of Industrial Information Integration* (2024): 100549.

[7] Labu, Md Rasheduzzaman, and Md Fahim Ahammed. "Next-Generation cyber threat detection and mitigation strategies: a focus on artificial intelligence and machine learning." *Journal of Computer Science and Technology Studies* 6.1 (2024): 179-188.

[8] Jimmy, F. N. U. "Cyber security Vulnerabilities and Remediation Through Cloud Security Tools." *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023* 2.1 (2024): 129-171.

[9] Hashmi, Ehtesham, Muhammad Mudassar Yamin, and Sule Yildirim Yayilgan. "Securing tomorrow: a comprehensive survey on the synergy of Artificial Intelligence and information security." *AI and Ethics* (2024): 1-19.

[10] Okoli, Ugochukwu Ikechukwu, et al. "Machine learning in cybersecurity: A review of threat detection and defense mechanisms." *World Journal of Advanced Research and Reviews* 21.1 (2024): 2286-2295.

[11] Atadoga, A., Sodiya, E. O., Umoga, U. J., & Amoo, O. O. (2024). A comprehensive review of machine learning's role in enhancing network security and threat detection. *World Journal of Advanced Research and Reviews*, *21*(2), 877-886.

[12] Sejfia, A., Das, S., Shafiq, S., & Medvidović, N. (2024, February). Toward Improved Deep Learning-based Vulnerability Detection. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering* (pp. 1-12).

[13] Lad, Sumit. "Harnessing Machine Learning for Advanced Threat Detection in Cybersecurity." *Innovative Computer Sciences Journal* 10.1 (2024).

[14] Jain, Vikas Kumar, and Meenakshi Tripathi. "An integrated deep learning model for Ethereum smart contract vulnerability detection." *International Journal of Information Security* 23.1 (2024): 557-575.

[15] Taofeek, Agboola Olayinka. "Development of a Novel Approach to Phishing Detection Using Machine Learning." *ATBU Journal of Science, Technology and Education* 12.2 (2024): 336-351.

[16] Alwahedi, F., Aldhaheri, A., Ferrag, M. A., Battah, A., & Tihanyi, N. (2024). Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. *Internet of Things and Cyber-Physical Systems*.

[17] Liang, P., Wu, Y., Xu, Z., Xiao, S., & Yuan, J. (2024). Enhancing Security in DevOps by Integrating Artificial Intelligence and Machine Learning. *Journal of Theory and Practice of Engineering Science*, *4*(02), 31-37.

[18] Regano, Leonardo, Daniele Canavese, and Luca Mannella. "A Privacy-Preserving Approach for Vulnerability Scanning Detection." *Proceedings of the Italian Conference on Cybersecurity (ITASEC 2024). CEUR-WS*. 2024.

[19] Arif, H., Kumar, A., Fahad, M., & Hussain, H. K. (2024). Future Horizons: AI-Enhanced Threat Detection in Cloud Environments: Unveiling Opportunities for Research. *International Journal of Multidisciplinary Sciences and Arts*, *3*(1), 242-251.

[20] Maddireddy, Bharath Reddy, and Bhargava Reddy Maddireddy. "Advancing Threat Detection: Utilizing Deep Learning Models for Enhanced Cybersecurity Protocols." *Revista Espanola de Documentacion Cientifica* 18.02 (2024): 325-355.

[21] Ceschin, F., Botacin, M., Bifet, A., Pfahringer, B., Oliveira, L. S., Gomes, H. M., & Grégio, A. (2024). Machine learning (in) security: A stream of problems. *Digital Threats: Research and Practice*, *5*(1), 1-32.

[22] Steenhoek, Benjamin, Hongyang Gao, and Wei Le. "Dataflow analysis-inspired deep learning for efficient vulnerability detection." *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*. 2024.

[23] Seas, C., Fitzpatrick, G., Hamilton, J. A., & Carlisle, M. C. (2024, January). Automated Vulnerability Detection in Source Code Using Deep Representation Learning. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0484-0490). IEEE.

[24] Mohammed, Ahmed. "The Web Technology and Cloud Computing Security based Machine Learning Algorithms for Detect DDOS Attacks." *Journal of Information Technology and Informatics* 3.1 (2024).

[25] Wang, R., Xu, S., Ji, X., Tian, Y., Gong, L., & Wang, K. (2024). An extensive study of the effects of different deep learning models on code vulnerability detection in Python code. *Automated Software Engineering*, *31*(1), 15.

[26] Reddy, Premkumar, Yemi Adetuwo, and Anil Kumar Jakkani. "Implementation of Machine Learning Techniques for Cloud Security in Detection of DDOS Attacks." *International Journal of Computer Engineering and Technology(IJCET)* 15.2 (2024).

[27] Sun, H., Cui, L., Li, L., Ding, Z., Li, S., Hao, Z., & Zhu, H. (2024). VDTriplet: Vulnerability detection with graph semantics using triplet model. *Computers & Security*, *139*, 103732.