Optimising Real-Time Threat Detection: A Hybrid SVM and ANN Approach

MSc Cybersecurity
Academic Internship

Chethanprasad Narasimhamurthy
Student ID: X22180591

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Chethanprasad Narasimhamurthy
Student ID:	X22180591
Programme:	MSc CyberSecurity **Year:** 2023-2024
Module:	Academic Internship
Supervisor:	Vikas Sahni
Submission Due Date:	14/12/2023
Project Title:	Optimising Real-Time Threat Detection: A Hybrid SVM and ANN Approach
Word Count:	7402 **Page Count** 21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Chethanprasad Narasimhamurthy

Date: 14/12/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimising Real-Time Threat Detection: A Hybrid SVM and ANN approach

Chethanprasad Narasimhamurthy
X22180591

Abstract

Real-time threat detection poses a significant challenge in the realm of cybersecurity. Recognizing abnormal activities requires advanced monitoring systems. This research introduces a hybrid threat detection model, merging Support Vector Machine (SVM) and Artificial Neural Network (ANN), to address the limitations of conventional monitoring systems. The model, designed for real-time threat detection, leverages SVM for feature extraction and ANN for pattern recognition, providing an innovative solution to evolving security landscapes.

Evaluations using UNSW-NB15[1] and NSL-KDD[2] datasets demonstrate the hybrid model's superior performance compared to a Logistic Regression model. The hybrid model exhibits higher accuracy of 94.83% for UNSW-NB15[1] and 95.36% for NSL-KDD[2] as compared to Logistic Regression model with accuracy of 93.85% for UNSW-NB15[1] and 94.48% for NSL-KDD[2] for contributing valuable benchmarks to intrusion detection methodologies. The SVM-ANN hybrid model, proven to be robust and adaptable, holds practical implications for effective intrusion detection. However, variations in execution times and the trade-off between false negatives and detection rates warrant further investigation.

Keywords: Threat Detection, SVM, ANN, Hybrid Approach, UNSW-NB15[1] Dataset and, NSL-KDD[2] Dataset

1 Introduction

In the realm of cybersecurity, the escalating of cyber threats has outpaced the efficacy of conventional defence mechanisms. Traditional rule-based and signature-based systems struggle to cope with the dynamic and evolving nature of contemporary attacks (Gander, et al., 2013). Traditional security measures, designed for on-premises environments, may not be fully equipped to handle the dynamic and distributed nature of cloud infrastructures (Aslan, et al., 2021). Consequently, there is a pressing need for adaptive and intelligent security mechanisms capable of dynamically responding to the ever-changing threat landscape (Sethi, et al., 2020).

The major challenge in cloud computing is the real-time detection of threats. Monitoring systems play a crucial role in recognizing abnormal activities (Abdelsalam, et al., 2021). In this context, machine learning (ML) has emerged as a transformative technology, promising to bolster security measures by proactively identifying and responding to evolving threats (Li, et al., 2013). The advent of machine learning has become pivotal in addressing these challenges, offering the potential for adaptive and intelligent threat detection.

[1] https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15
[2] https://www.kaggle.com/datasets/kiranmahesh/nslkdd

Understanding the landscape of security threats is crucial for developing effective defence mechanisms. The shared responsibility model, which delineates security responsibilities between cloud service providers and customers, introduces complexities in securing infrastructures. Threats may target vulnerabilities in cloud configurations, insecure application programming interfaces (APIs), or exploit misconfigurations in the deployment (Meryem & Ouahidi, 2020).

The evolution of threats necessitates advanced and adaptive security mechanisms, and ML algorithms offer a promising avenue for addressing these challenges (Soni & Kumar, 2022). ML algorithms can be applied to various aspects of cloud security, including anomaly detection, behavioral analysis, and pattern recognition. These algorithms excel at processing and analysing vast datasets, learning from historical data, and identifying patterns indicative of potential security threats (Kumar, et al., 2022). The integration of ML into cloud security frameworks aims to enhance the ability to detect and respond to threats in real-time, ultimately strengthening the overall resilience of cloud infrastructures (Saranya, et al., 2020).

The importance of effective threat detection cannot be overstated, as the consequences of cyber-attacks extend beyond individual systems to impact national security, financial stability, and personal privacy (Ou, 2019). The sheer volume and sophistication of cyber threats necessitate advanced, adaptive, and efficient detection mechanisms. This research addresses the critical need for robust cybersecurity defences by exploring the synergies between SVM and ANN, aiming to optimise real-time threat detection in the face of evolving digital risks.

1.1 Research question

1. How does the hybrid approach combining SVM and ANN optimise the efficacy of real-time threat detection?
2. What is the effectiveness and reliability of this model when evaluated using the UNSW-NB15[1] and NSL-KDD[2] datasets?

This research is driven by the shortcomings of conventional rule-based threat detection systems and addresses the need for advanced threat detection model by proposing a hybrid approach that combines SVM and ANN. The hybrid model aims to harness the strengths of both algorithms, creating a synergistic system capable of robustly identifying and classifying security threats. SVM, a powerful supervised learning algorithm, serves as the initial detector to extract distinctive features from the data.

The extracted features from SVM are then fed into an ANN, a neural network-inspired model known for its ability to handle complex, non-linear relationships within data. The ANN enhances the model's capacity to discern intricate patterns indicative of sophisticated threats, contributing to a more comprehensive and adaptive threat detection system. The integration of SVM and ANN creates a hybrid model that addresses the limitations of individual algorithms, providing a robust solution to the dynamic threat landscape of cloud computing.

The primary objectives of this research are as follows:

1. **Hybrid Model Development:** Developing a hybrid threat detection model that seamlessly integrates SVM and ANN to enhance the accuracy and effectiveness of threat identification in cloud monitoring systems.

2. **Preprocessing Techniques:** Implementation of tailored preprocessing techniques, including robust scaling and label encoding, to ensure the model's adaptability to diverse datasets commonly encountered in security scenarios.
3. **Evaluation:** Evaluating the hybrid model's performance using a range of metrics, including accuracy, precision, recall, false-positive rates, and detection rates.
4. **Visualization Tools:** Use of visualization tools such as ROC curves and Precision-Recall curves to facilitate the interpretation of model outcomes, offering insights into its performance characteristics.

This research contributes to the field of cloud security by proposing an innovative and adaptive approach to threat detection. The hybrid model, combining SVM and ANN, addresses the limitations of individual algorithms, providing a robust solution to the dynamic threat landscape of cloud computing. By leveraging advanced ML techniques, this research aims to fortify the proactive identification of security threats, reducing response times and mitigating potential damages.

Furthermore, the study emphasizes the importance of tailored preprocessing techniques in the context of cloud security. The integration of robust scaling and label encoding ensures that the hybrid model can effectively handle the diverse datasets encountered in real-world cloud environments, where features may include both categorical and numerical components.

While this research endeavours to contribute to the advancement of real-time threat detection, it is important to acknowledge certain limitations inherent in the study. The performance of the hybrid model may be influenced by the specific characteristics of the datasets used for evaluation, and the generalization of findings to diverse cybersecurity environments requires cautious consideration. Additionally, the complexity of cyber threats may extend beyond the scope of the selected algorithms, warranting ongoing exploration of emerging techniques to address evolving challenges.

1.2 Report Structure

The subsequent portion of the document is organized as follows. In Section 2, a comparison and assessment are conducted between the connected literature review and the content of this paper. Section 3 outlines the intended methodology for the research, along with the expected/forthcoming research steps to conclude the project. The description of experiments, data processing, and presentation of results are detailed in Section 4. Lastly, Section 5 encompasses the conclusion and outlines future work.

2 Related Work

2.1 Related works based on algorithms

In the domain of cloud security, (Du, et al., 2020) introduced the Edge of Things (EoT) algorithm, designed for detecting web attacks within cloud platforms. The novelty of EoT lies in its deployment of simultaneous deep models on edge devices, enhancing system stability and enabling efficient updates. The algorithm demonstrated remarkable performance, achieving an accuracy of 99.410%, a true positive rate of 98.91%, and a detection rate of 99.55%. To further improve accuracy, the authors recommended integrating additional models

such as hidden decision trees, Markov models, and long short-term memory (LSTM) deep learning models.

(Delplace, et al., 2020) focused on leveraging machine learning to categorize harmful traffic within a network. Their study conducted a thorough analysis of NetFlow datasets, identifying 22 pertinent features directly related to the issue. Among the tested machine learning algorithms, the random forest classifier emerged as the most successful, effectively identifying over 95% of botnets in various in scenarios. Although the authors attempted to improve accuracy by employing a bootstrap method to supplement the data, they encountered challenges in achieving substantial enhancements. They suggested the use of techniques such as recursive deep neural networks as potential avenues for addressing this difficulty.

(Alzahrani & Alenazi, 2021) explores the integration of software-defined networking (SDN) and ML algorithms for network intrusion detection, addressing the vulnerabilities introduced by SDN's enhanced flexibility. The study leverages classical and advanced tree-based ML techniques, namely decision tree, random forest, and XGBoost, to detect malicious behavior in the network. Utilizing the NSL-KDD[2] dataset, the proposed methods achieved an impressive 95.95% accuracy in a multi-class classification task. The conclusion highlights the growing interest in SDN-based machine learning algorithms, emphasizing the significance of feature normalization, selection, and data preprocessing in optimizing algorithm performance. The proposed XGBoost model outperformed seven other algorithms, showcasing its effectiveness in real-time attack detection. The future work plan includes implementing deep neural network algorithms and comparing them to enhance anomaly detection efficiency in NIDS.

(Thilagam & Aruna, 2021) proposed an advanced Intrusion Detection System. It leverages an optimised Recurrent Convolutional Neural Network (RC-NN) in tandem with the Ant Lion optimization algorithm. This approach hybridizes Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), efficiently classifying attacks within the cloud network layer.

Experimental results demonstrate the optimised custom RC-NN-IDS model's impressive performance, achieving a 94% classification accuracy and a reduced error rate of 0.0012. Superior metrics, including True Positive Rate (TPR) and True Negative Rate (TNR), underscore its effectiveness compared to existing classifiers.

The study concludes by highlighting the proposed approach's potential for future extensions, envisioning a management module for initiating preventive actions post-intrusion detection based on classified results, enhancing the overall security posture of cloud networks.

(Kumar, et al., 2022) addressed the evolving threat landscape posed by developments in IoT, Cloud Infrastructures, and sectors like E-Commerce, Banking, and Healthcare. Recognizing intrusion detection as pivotal, the authors introduce a novel Fuzzy Min Max Neural Networks-Based Intrusion Detection System (FMMNN-IDS) using a fuzzy min-max learning algorithm.

The FMMNN-IDS model demonstrates superior accuracy in both binary and multiclass classification, outperforming state-of-the-art approaches outlined in the literature.

Notably, it enhances intrusion detection accuracy, achieving a commendable balance between identification rate and a low false positive rate, particularly in multiclass classification using the NSL-KDD dataset. The paper concludes by emphasizing the significance of fuzzy

logic and neural networks in enhancing the proposed FMMNN-IDS's performance. Future research avenues include optimizing FMMNN algorithm training time and classification efficacy in the realm of intrusion prevention.

2.2 Related work based on training and performance

(Chen, et al., 2020) conducted a comparative analysis of the efficiency of various ML algorithms, including decision tree, deep belief network, and SVM, in discerning spam, intrusion, and malware. While highlighting the effectiveness of these algorithms in threat detection, the study revealed the absence of a universally effective approach due to the scarcity of comprehensive datasets for thorough testing.

(Bera, et al., 2020) addressed the duration of algorithm training in the context of a malware detection system. Their approach, centered on clustering and trend micro locality-sensitive hashing (TLSH), employed ML techniques such as random forests, decision trees and, logistic regression. The outcomes showcased improved classification accuracy and a reduced false positive rate, all while minimizing the training time for the algorithms.

(Asif, et al., 2022) address the escalating need for cybersecurity in the context of the Internet of Things (IoT) and the massive expansion of computer networks. The study proposes a MapReduce-Based Intelligent Model for Intrusion Detection (MR-IMID), combining clustering techniques with ML to intelligently automate intrusion detection. MR-IMID processes large datasets efficiently using commodity hardware, utilizing multiple network sources in real-time for intrusion detection. The model predicts unknown test scenarios, storing data in the database for future consistency.

During validation and training MR-IMID achieves a detection accuracy of 95.7%, and 97.7% respectively, surpassing previous approaches. The paper underscores the effectiveness of combining MapReduce and ML techniques, specifically ANN, in parallel clustering and feature extraction. The proposed MR-IMID proves scalable and robust, demonstrating its potential as an advanced intrusion detection system.

2.3 Related work based on approach

(Rabbani, et al., 2020) introduced a groundbreaking Particle Swarm Optimization-based Probabilistic Neural Network (PSO-PNN) for enhancing security in cloud-based environments. Validated with the UNSW-NB15[1] dataset, the system exhibited promising results in characterizing diverse malicious behaviors. Visual representations highlighted its proficiency in distinguishing between normal and malicious activities, with high accuracy rates in classifying modern attacks. The collaborative PSO-PNN system showcased effectiveness in addressing security challenges. Acknowledging the importance of feature extraction, the authors proposed future work involving deep learning techniques for an ideal recognition system.

(Singh & Khare, 2021) highlight the necessity for efficient and updated systems. This paper advocates for the implementation of NIDS using ML techniques and up-to-date intrusion datasets to ensure effective modeling. The article provides a comprehensive overview of publicly available labeled intrusion datasets and ML techniques, delving into literary works applying ML in various networking scenarios, including traditional networks, WSNs, Ad-Hoc, cloud networks, and IoT networks.

It emphasizes the critical role of ML techniques in handling complex data while discussing their characteristics and limitations. The study scrutinizes recent NIDS models that leverage ML techniques and public intrusion datasets across diverse networking environments. By elucidating current security challenges, solutions, outcomes, and future directions, the paper serves as a valuable resource for researchers seeking to enhance existing NIDS models and develop new effective ones.

(Abdelsalam, et al., 2021) proposed a method for detecting malware online by utilizing performance metrics at the process level. Their study employed various ML models, including the support vector classifier, K-nearest neighbor, random forest classifier, Gaussian Naive Bayes, gradient boosted classifier, and Convolutional Neural Networks. With a dataset comprising large malicious samples, the DenseNet-121 (CNN) deep learning model exhibited the most effective performance for identifying malware in real-time within cloud IaaS.

(Aldallal & Alisa, 2021) addressed the limitations of current intrusion detection systems, specifically their susceptibility to false alarms. They proposed a hybrid approach incorporating ML techniques such as SVM and genetic algorithms. The testing, conducted using the CICIDS2017 dataset and benchmark datasets NSL-KDD CUP 99 and NSL-KDD[2], demonstrated substantial accuracy improvements up to 5.74% across different datasets.

(Aslan, et al., 2021) introduced the concept of behavior-based detection through the proposed Common Behavior and Characteristic Malware (CBCM). The study outlined two primary phases: the client phase, where suspicious samples are transmitted to the cloud for analysis, and the cloud environment phase, encompassing sample analysis, behavior-based detection, and classification of samples as either malware or benign.

The achieved outcomes demonstrate a high detection rate of 99.8%, a minimal false positive rate of 0.4%, and an overall accuracy of 99.7% across large test samples.

Additionally, (Ahsan, et al., 2022) provided a critical examination of machine learning models' limitations in the context of Intrusion Detection Systems. The survey explored diverse machine learning approaches for identifying various types of attacks and suggested the exploration of emerging techniques such as Homomorphic Encryption, along with the assessment of threats related to quantum computing.

2.4 Related work based on threat

(Rana, et al., 2022) conducted research focused on creating an intrusion detection system specialized in identifying zero-day attacks. Testing with the NSL-KDD[2] and UNSW-NB15[1] datasets demonstrated the effectiveness of the FCM-ANN approach, particularly in the case of the UNSW-NB15[1] dataset.

(Kaushik, et al., 2022) employed the preprocessed NSL-KDD[2] dataset to implement a network intrusion detection system in the cloud. This approach guaranteed high availability and efficiency in threat monitoring.

The study underscored the importance of preprocessing in enabling ML algorithms to improve accuracy and eliminate the need for ongoing manual supervision.

(SaiSindhuTheja & Shyam, 2021) addressed the critical challenge of detecting Denial of Service (DoS) attacks in cloud computing. Their proposed system, leveraging the Oppositional Crow Search Algorithm (OCSA) and Recurrent Neural Network (RNN) classifier, showcased

superior performance. The two-stage process involved OCSA for feature selection and RNN for classification, ensuring the identification of standard and compromised data.

Experimental results, using a benchmark dataset, demonstrated the technique's excellence, outperforming conventional methods by significant margins in precision, recall, F-measure, and accuracy. Future work is envisioned to include an attack prevention system for cloud-based systems, exploring cross-validation, real-time identification, and prevention of DoS attacks in a multi-cloud environment.

Table 1: Literature review

Reference	Year	Areas Focused	ML Techniques	Issues Addressed
(Delplace, et al., 2020)	2020	Network Traffic Categorization	Random Forest Classifier, Machine Learning on NetFlow Datasets	Botnet Identification, Harmful Traffic Categorization, Feature Analysis
(Du, et al., 2020)	2020	Web Attack Detection, Cloud Security	Simultaneous Deep Models on Edge Devices	System Stability, Efficient Updates
(Bera, et al., 2020)	2020	Malware Detection System Training Duration	Clustering, TLSH, Decision Trees, Random Forests, Logistic Regression	Classification False Positive Rate, Accuracy, Reduced Training Time
(Chen, et al., 2020)	2020	ML Algorithm Efficiency, Threat Detection	Deep Belief Network, Decision Tree, SVM	Scarcity of Comprehensive Datasets
(Rabbani, et al., 2020)	2020	PSO-PNN for Security in Cloud Environments	Particle Swarm Optimization, Probabilistic Neural Network	Distinguishing Malicious Behaviors, Visual Representations
(Abdelsalam, et al., 2021)	2021	Online Malware Detection, Process-Level Metrics	SVC, RFC, CNN (DenseNet-121)	Process-Level Performance Metrics, Real-time Detection
(Aldallal & Alisa, 2021)	2021	Intrusion Detection Systems, False Alarms	Hybrid Approach (SVM, Genetic algorithms)	False Alarms, Accuracy Improvement
(Alzahrani & Alenazi, 2021)	2021	SDN-based Network Intrusion Detection	Decision Tree, Random Forest, XGBoost	Malicious Behavior Detection, Significance of Feature Normalization
(Aslan, et al., 2021)	2021	Behavior-Based	Various ML Models (SVM,	Malware Detection, False Positive Rate

		Detection, CBCM	Random Forest, K-NN, Gradient Boosted, GNB, CNN)	
(SaiSindhuTheja & Shyam, 2021)	2021	DoS Attack Detection in Cloud Computing	OCSA and RNN Classifier	Superior Performance, Two-Stage Process, Future Work on Attack Prevention
(Singh & Khare, 2021)	2021	Necessity for Efficient and Updated NIDS Systems	ML Techniques for NIDS, Overview of Public Datasets	Handling Complex Data, Characteristics and Limitations of ML Techniques
(Thilagam & Aruna, 2021)	2021	Advanced Intrusion Detection System	Optimised RC-NN with Ant Lion Optimization Algorithm	Efficient Attack Classification, True Positive Rate, True Negative Rate
(Ahsan, et al., 2022)	2022	Limitations of ML Models in IDS, Emerging Techniques	Various ML Approaches, Homomorphic Encryption, Threat Assessment	Identifying Various Attack Types, Threats Related to Quantum Computing
(Asif, et al., 2022)	2022	MapReduce-Based Intelligent Model for IDS	Clustering, Machine Learning (ML), MapReduce	Efficient Intrusion Detection, Real-time Processing of Large Datasets
(Kaushik, et al., 2022)	2022	Network Intrusion Detection, Cloud	Preprocessed NSL-KDD[2] Dataset, Emphasis on Preprocessing	High Availability, Efficiency in Threat Monitoring, Preprocessing for Accuracy Improvement
(Kumar, et al., 2022)	2022	Fuzzy Min Max Neural Networks-Based IDS	Fuzzy Min-Max Learning Algorithm	Intrusion Detection in IoT, Cloud Infrastructures, and Sectors like E-Commerce
(Rana, et al., 2022)	2022	Zero-Day Attack Detection	FCM-ANN	Zero-Day Attacks, Testing with NSL-KDD[2] and UNSW-NB15[1] Datasets

While several papers propose hybrid approaches, there is no explicit mention of integrating different hybrid models. Addressing how different hybrid models could enhance threat detection efficacy could be explored. There is a lack of direct comparison between different models using common datasets, such as NSL-KDD[2] and UNSW-NB15[1].

In essence, the research questions were framed to bridge existing gaps. The lack of detailed exploration into the synergies of SVM-ANN hybrid models and the absence of evaluations on common datasets. By addressing these questions, the research aims to contribute valuable insights into the practical effectiveness of the SVM-ANN hybrid approach for real-time threat detection in cloud environments.

3 Research Methodology

The proposal involves several steps outlined below.

1. **Data Collection and Preprocessing:** This project utilises NSL-KDD[2] and UNSW-NB15[1] datasets, with the NSL-KDD[2] dataset containing labelled network traffic data, and the more recent and comprehensive UNSW-NB15[1] dataset. The collected dataset is divided into training data of 80%, and testing data of 20% subsets. The training subset is employed to train the ML algorithm.
2. **Preprocessing the Dataset:** Data cleaning identifies and eliminates duplicates, while normalization ensures uniformity within the dataset. Feature selection involves choosing relevant features from the cleaned data to train the algorithm.
3. **Model Training and Optimization:** The SVM-ANN algorithm is trained using the NSL-KDD[2] and UNSW-NB15[1] datasets. Optimization techniques such as stacking, bagging, and recursive deep neural networks are employed to enhance accuracy. Additionally, the model addresses imbalanced data, as threats are inherently unpredictable.
4. **Evaluation:** Testing involves evaluating the performance of the designed model using metrics accuracy, recall, precision, false positive rate, false negative rate, F1-score, and detection rate, utilizing the datasets. Furthermore, the results are compared with those of existing methods to assess the extent of improvements achieved.

4 Design Specification

The design specification delineates the architecture and functionality of the hybrid model, which combines a SVM and an ANN. It also describes the preprocessing steps and evaluation metrics.

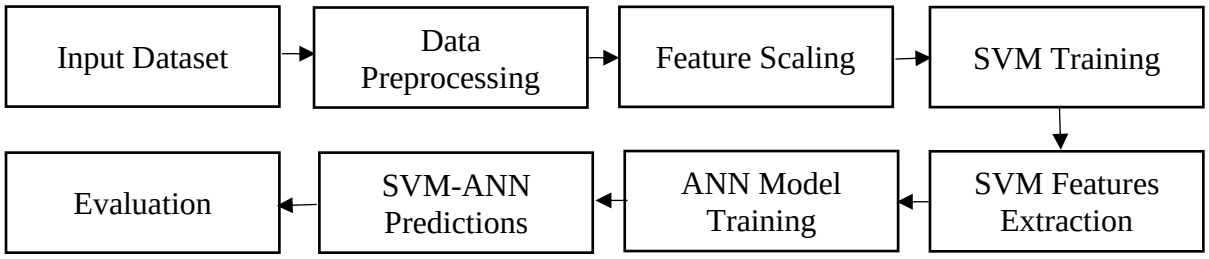

Figure 1: SVM-ANN hybrid model's architecture.

Figure 1 shows the architecture of SVM-ANN model and the processing begins with the input dataset, where raw network activity data is provided. This data undergoes data preprocessing, involving cleaning and organizing to prepare it for analysis. Subsequently, feature scaling standardizes numerical features, ensuring uniformity for models sensitive to input magnitude.

The SVM Training block employs SVM to train a model for pattern recognition in the pre-processed data. SVM features extraction extracts relevant features from the SVM-trained model, serving as inputs for the ANN model training stage. The SVM-ANN predictions block combines outputs from both SVM and ANN, providing the final predictions. The evaluation phase assesses the hybrid model's performance.

4.1 Hybrid Model Architecture

4.1.1 SVM Component Functionality

The SVM component serves as the initial phase of the hybrid model. It utilizes a linear kernel for binary classification. The linear kernel is chosen for its simplicity and effectiveness in linearly separable datasets. The SVM's decision function is employed to extract features from the input data. This function calculates the distance of each data point from the decision boundary, generating a set of numerical values representing the SVM-extracted features.

4.1.2 ANN Component Functionality

The ANN component constitutes the second phase of the hybrid model, building upon the features extracted by the SVM. It consists of two hidden layers with 100 and 50 neurons, respectively. The choice of hidden layer sizes is based on empirical testing and aims to capture complex patterns in the data.

Rectified Linear Unit (ReLU) activation functions are employed for the hidden layers, introducing non-linearity to the model. This allows the ANN to learn intricate relationships within the data.

4.1.3 Data Preprocessing

To ensure the robustness of the model against outliers, numerical feature scaling is applied using the RobustScaler. This method scales features by removing the median and scaling data according to the interquartile range, making the model less sensitive to extreme values.

Categorical features are encoded using one-hot encoding, enhancing the model's ability to interpret categorical information. The drop_first parameter is set to True during one-hot encoding to prevent multicollinearity, where one category can be inferred from the others.

The target column undergoes transformation for binary classification. Instances labelled as "normal" are assigned the value 0, while other instances are assigned the value 1. This transformation facilitates a binary classification task, with the goal of distinguishing normal instances from those indicative of an attack.

4.2 Evaluation Metrices

Accuracy: It is proportion of correctly classified instances among the total instances It is calculated using true positives (TP), true negatives (TN), false positives (FP), false negatives (FP) and give as below.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Precision: It is the ratio of true positive predictions to the total predicted positives. It measures the model's ability to avoid false positive predictions, indicating the precision of positive classifications.

$$Precision = \frac{TP}{(TP + FP)}$$

Recall: It is the ratio of true positive predictions to the total actual positives. It gauges the model's ability to capture all positive instances, providing insight into its sensitivity to positive events.

$$Recall = \frac{TP}{(TP + FN)}$$

F1 score: It is the harmonic mean of precision and recall. It balances the trade-off between precision and recall, offering a single metric that considers both false positives and false negatives.

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

Confusion matrix: It is a table that summarizes the model's performance, showing counts of true positive, true negative, false positive, and false negative predictions. It provides a detailed breakdown of classification results.

False positive rate: It is the ratio of false positive predictions to the total actual negatives. It quantifies the rate of falsely identified positive instances among all actual negatives.

$$False\ Postive\ Rate = \frac{FP}{(FP + TN)}$$

False negative rate: It is the ratio of false negative predictions to the total actual positives. It measures the rate of instances incorrectly classified as negative among all actual positives.

$$False\ Negative\ Rate = \frac{FN}{(FP + TN)}$$

Detection rate: It is the ratio of true positive predictions to the total actual positives. It signifies the model's effectiveness in correctly identifying positive instances.

$$Detection\ Rate = \frac{TP}{(TP + FN)}$$

5 Implementation

The implementation involves the creation of a hybrid model for intrusion detection, combining a SVM and ANN. The goal is to train and evaluate hybrid model and compare its performance with Logistic Regression classifiers for intrusion detection. This script is designed to process a dataset, perform necessary preprocessing steps, train the hybrid model, and evaluate its performance.

5.1 Hybrid Model Architecture:

The hybrid model consists of two main components – the SVM and the ANN. The SVM serves as the initial phase, employing a linear kernel for binary classification. The decision function of the SVM extracts features from the input data, calculating the distance of each point from the decision boundary. These features are then utilized by the ANN component, which includes two hidden layers with 100 and 50 neurons, respectively. The choice of hidden layer sizes is based on empirical testing, aiming to capture complex patterns in the data. Rectified Linear Unit (ReLU) activation functions introduce non-linearity to the model, allowing the ANN to

learn intricate relationships within the data. The user is prompted to input the file path of the dataset, ensuring flexibility in dataset selection. The following steps provide an overview of the process:

5.1.1 Data Loading and Preprocessing:

The user is prompted to input the path to the dataset, which is to be a CSV file. The dataset is loaded into a Pandas DataFrame ('data_train'). Descriptive statistics and information about the dataset are displayed, aiding in initial exploration. The 'preprocess' function is called to handle data cleaning and encoding based on the column names ('labels' or 'label').

5.1.2 Data Transformation:

Categorical features are processed, and label encoding is applied to the target variable ('labels' or 'label'). Pie charts are generated to visualize the distribution of categorical features, providing insights into the dataset's composition. Numerical features are standardized using the 'StandardScaler' to ensure uniformity and improve model performance. One-hot encoding is applied to categorical features, creating dummy variables.

5.1.3 Model Training:

The dataset is split into training and testing sets using the 'train_test_split' function. Non-numeric columns are removed from the input features, as they are not suitable for some models. Standard scaling is applied to the numeric features of the training and testing sets. Three models are trained: SVM with a linear kernel is trained on the scaled training data. The decision function output of the SVM on the training set is used as a feature for training an ANN. Logistic Regression is trained on the scaled training data.

5.1.4 Model Evaluation:

The trained models are evaluated using the testing set. For each model, accuracy, precision, recall, F1 score, false positive rate, false negative rate, and detection rate are computed and printed.

5.1.5 Results Visualization:

Bar plots are created for each model, displaying performance metrics such as accuracy, precision, recall, F1 score, false positive rate, false negative rate, and detection rate. These visualizations help compare the performance of the SVM-ANN hybrid model and the Logistic Regression model.

5.2 Tools and Libraries

The implementation is carried out in Python, a versatile programming language widely used in the field of machine learning. Several libraries are employed to streamline the implementation. Scikit-learn is utilized for implementing the SVM model, preprocessing steps, and calculating evaluation metrics. Pandas is employed for efficient data manipulation and analysis. Matplotlib and Seaborn are used for creating visualizations to aid in result interpretation, and NumPy is utilised for numerical operations and array manipulations. The details are given in configuration manual.

6 Evaluation

To assess the performance of the SVM-ANN hybrid model, experiments were conducted by comparing it against an existing linear regression model using two benchmark datasets: the NSL-KDD[2] dataset and the UNSWNB15 dataset. The analysis of the results involves an examination of seven performance metrics, namely accuracy, precision, recall, F1 score, false positive rate, false negative rate, and detection rate.

6.1 Experiment 1: Performance of the model using UNSW-NB15[1] dataset

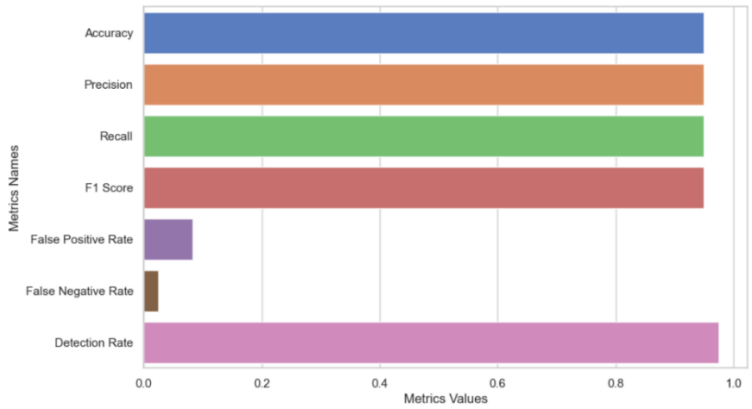

Figure 2: Performance of SVM-ANN model when tested using UNSW-NB15[1] dataset

The SVM-ANN hybrid model exhibits commendable performance against UNSW-NB15[1] dataset, boasting an impressive accuracy of 94.84%. This underscores its proficiency in correctly classifying instances, as evidenced by the precision of 94.89% and recall of 94.84%. The balance between precision and recall is further validated by the F1 score of 94.83%, attesting to the model's robustness in binary classification tasks. Notably, the false positive rate is low at 7.99%, signifying minimal misclassification of actual negatives, while the false negative rate stands at 2.84%, indicating a high sensitivity to positive instances.

The detection rate, a critical metric in scenarios prioritizing the identification of positive cases, is notably high at 97.16%. Furthermore, the model's computational efficiency is evidenced by a total execution time of 168.41 seconds.

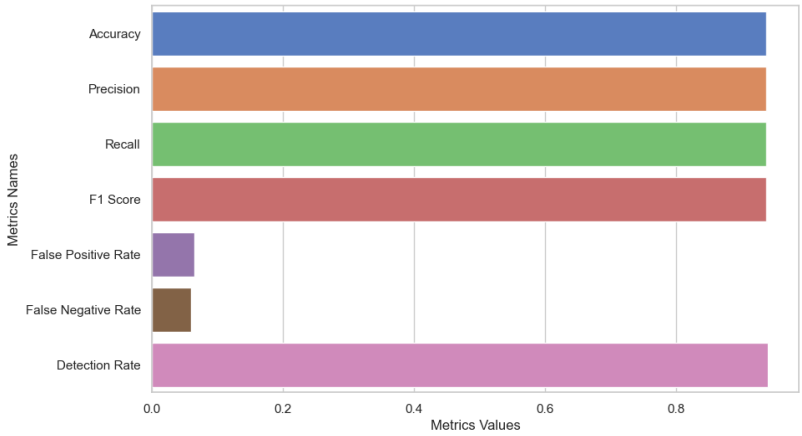

Figure 3: Performance of Logistic Regression model when tested using UNSW-NB15[1] dataset

The Logistic Regression model in this case exhibits a commendable performance with an accuracy of 93.85%. Precision, recall, and F1 score metrics are all at high levels, specifically 93.85%, 93.85%, and 93.85%, respectively. These metrics indicate the model's proficiency in

making accurate positive predictions while effectively capturing true positive instances. The false positive rate is relatively low at 6.44%, suggesting a moderate rate of misclassifying actual negatives. The false negative rate stands at 5.91%, indicating a good sensitivity to positive instances. The detection rate, measuring the model's ability to correctly identify positive cases, is substantial at 94.09%.

Table 2: Model's performance comparison using UNSW-NB15[1]

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	False Positive rate (%)	False Negative rate (%)	Detection rate (%)
SVM-ANN model	94.83	94.88	94.83	94.82	7.94	2.90	97.10
Logistic Regression Model	93.85	93.85	93.85	93.85	6.44	5.91	94.09

The comparison between the SVM-ANN Hybrid model and the Logistic Regression model reveals differences in their performance on UNSW-NB15[1] dataset. The SVM-ANN Hybrid model exhibits slightly higher accuracy at 94.83%, surpassing the Logistic Regression model's accuracy of 93.85%. Precision is also marginally higher for the SVM-ANN Hybrid model at 94.88%, compared to the Logistic Regression model's precision of 93.85%.

While both models demonstrate similar recall values (94.83% for SVM-ANN Hybrid and 93.85% for Logistic Regression), indicating their ability to capture relevant instances, the SVM-ANN Hybrid model achieves a slightly better balance between precision and recall as reflected in its F1 Score of 94.82%, in contrast to the Logistic Regression model's F1 Score of 93.85%. Notably, the SVM-ANN Hybrid model outperforms in terms of the detection rate at 97.10%, emphasizing its capability to effectively identify positive instances. However, the Logistic Regression model excels in terms of a lower false positive rate (6.44%) compared to the SVM-ANN Hybrid model (7.94%), highlighting its proficiency in avoiding misclassifying negative instances.

6.2 Experiment 2: Performance of the model using NSL-KDD[2] dataset

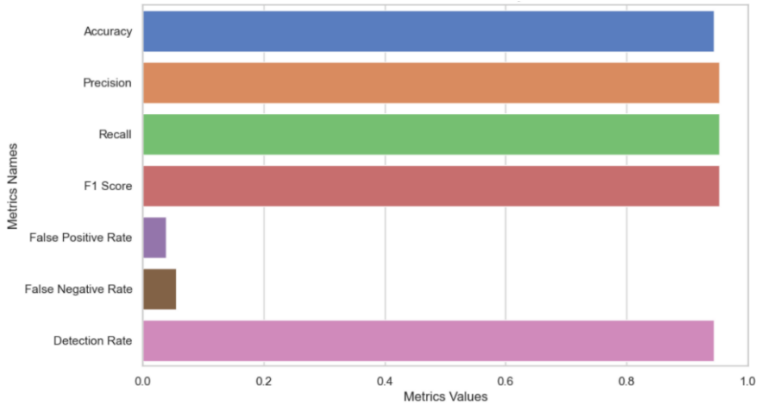

Figure 4: Performance of SVM-ANN model when tested using NSL-KDD[2] dataset

The SVM-ANN hybrid model showcases remarkable performance against NSL-KDD² dataset with an accuracy of 95.27%. This high accuracy is supported by precision and recall metrics of 95.29% and 95.27%, respectively, demonstrating the model's ability to make accurate positive predictions while effectively capturing true positive instances. The balanced F1 score of 95.26% further reinforces the model's overall effectiveness in binary classification tasks. The false positive rate is impressively low at 3.15%, indicating a minimal rate of misclassifying actual negatives, while the false negative rate stands at 6.53%, illustrating a strong sensitivity to positive instances. The detection rate, a critical measure of the model's ability to correctly identify positive cases, is substantial at 93.47%.

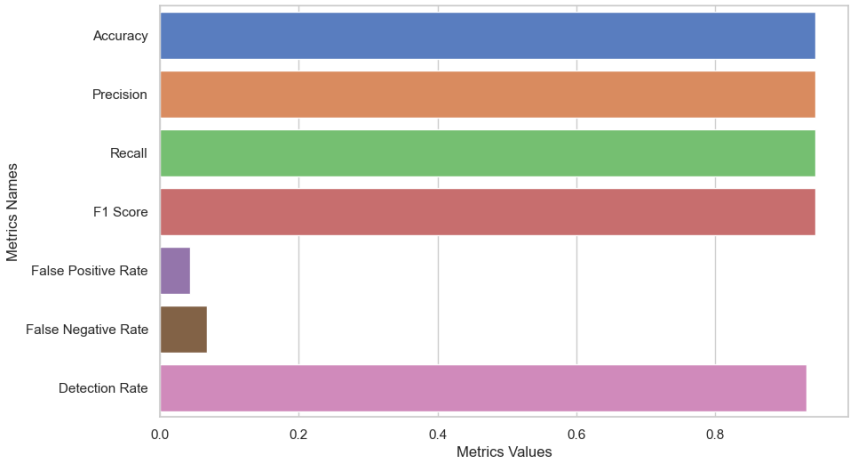

Figure 5: Performance of Logistic Regression model when tested using NSL-KDD² dataset

The Logistic Regression model demonstrates a strong performance on the given dataset, achieving an impressive accuracy of 94.48%. Precision, recall, and F1 score metrics all exhibit excellent values of 94.49%, 94.48%, and 94.48%, respectively. These metrics indicate the model's ability to accurately predict positive instances while effectively capturing true positives. The false positive rate is relatively low at 4.37%, suggesting a minimal rate of misclassifying actual negatives. The false negative rate stands at 6.83%, indicating a good sensitivity to positive instances. The detection rate, measuring the model's ability to correctly identify positive cases, is high at 93.17%.

Table 2: Model's performance comparison using NSL-KDD²

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	False Positive rate (%)	False Negative rate (%)	Detection rate (%)
SVM-ANN model	95.36	95.37	95.36	95.35	3.46	5.99	94.00
Logistic Regression Model	94.48	94.49	94.48	94.48	4.37	6.83	93.17

The comparison between the SVM-ANN Hybrid model and the Logistic Regression model reveals differences in their performance on NSL-KDD² dataset. The SVM-ANN Hybrid model achieves a commendable accuracy of 95.36%, surpassing the Logistic Regression model's accuracy of 94.48%. Precision, a measure of positive prediction accuracy, is slightly higher for

the SVM-ANN Hybrid model at 95.37%, compared to the Logistic Regression model's precision of 94.49%. Both models demonstrate high recall values (95.36% for SVM-ANN Hybrid and 94.48% for Logistic Regression), indicating their effectiveness in capturing relevant instances.

The F1 Score, representing the harmonic mean of precision and recall, is also marginally higher for the SVM-ANN Hybrid model at 95.35%, highlighting a balanced performance. Notably, the SVM-ANN Hybrid model achieves a lower false positive rate (3.46%) compared to the Logistic Regression model (4.37%), showcasing its proficiency in avoiding misclassifying negative instances. However, the Logistic Regression model has a lower false negative rate (6.83%) compared to the SVM-ANN Hybrid model (5.99%), emphasizing its ability to minimize the misclassification of positive instances. The detection rate is higher for the SVM-ANN Hybrid model at 94.00%, indicating its superior capability to identify positive instances.

6.3 Discussion

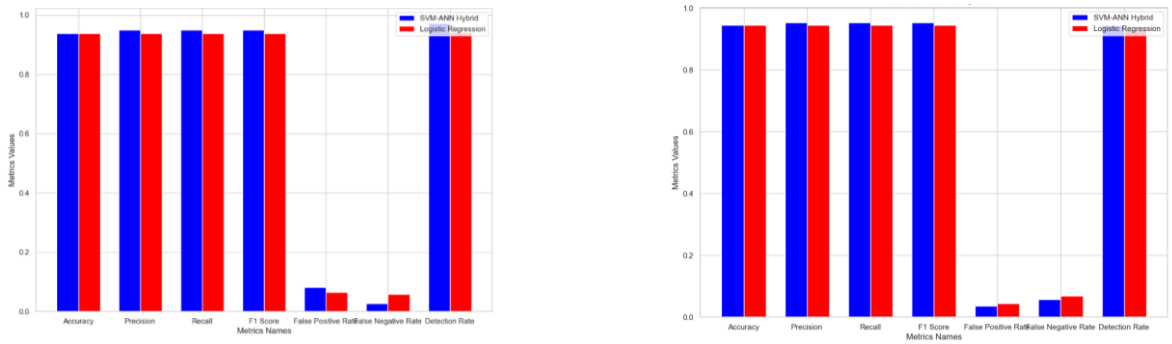

Figure 6: Performance comparison of SVM-ANN and Logistic Regression with UNSW-NB15[1] and NSL-KDD[2] datasets respectively

Performance Metrics Consistency: Across the two experiments which were carried out against the different datasets, the consistency of performance metrics, including accuracy, precision, recall, F1 score, false positive rate, false negative rate, and detection rate, is noteworthy. This suggests the reliability and stability of the SVM-ANN hybrid model across various scenarios.

Execution Time Variation: The noticeable variation in execution times across experiments, ranging from 168.41 seconds to 404.28 seconds, warrants attention. This could be indicative of potential computational complexities or resource constraints in certain cases. A deeper analysis of the reasons behind these variations is essential to optimise the model's efficiency for real-time applications. Consideration of parallel processing or optimization algorithms may be explored for potential improvements.

False Negative Rate and Detection Rate Trade-off: The trade-off between the false negative rate and the detection rate is a critical aspect. While achieving a high detection rate is desirable, it comes at the cost of a higher false negative rate in some experiments. This trade-off should be carefully considered based on the specific application requirements. Further research may explore techniques to fine-tune the model parameters to achieve a more balanced trade-off or consider domain-specific adjustments.

The experiments evaluating the SVM-ANN hybrid model in the domain of cloud security align with and contribute to existing literature, providing a nuanced understanding of the

model's strengths and potential areas for refinement. Notably, (Du, et al., 2020)'s EoT algorithm demonstrated exceptional accuracy in detecting web attacks within cloud platforms, suggesting potential avenues for enhancing the SVM-ANN hybrid model's accuracy by integrating additional models, as recommended by (Du, et al., 2020).

(Delplace, et al., 2020) focus on machine learning for network traffic categorization, particularly the success of the random forest classifier, offers a benchmark for the SVM-ANN hybrid model's notable accuracy (95.27%) and suggests exploring techniques like recursive deep neural networks to address challenges in improving accuracy.

(Chen, et al., 2020) comparative analysis of machine learning algorithms highlights the effectiveness of diverse models in threat detection, resonating with the SVM-ANN hybrid approach's use of various models. (Bera, et al., 2020) emphasis on minimizing algorithm training time aligns with the SVM-ANN hybrid model's clustering and locality-sensitive hashing focus, both achieving improved classification accuracy and reduced false positive rates. (Aldallal & Alisa, 2021) hybrid intrusion detection system, addressing false alarms, prompts consideration for enhancing the SVM-ANN hybrid model's accuracy and robustness through similar hybrid approaches.

(Ahsan, et al., 2022) behavior-based detection through CBCM and (Aldallal & Alisa, 2021) proposed approach showcase robust performance metrics, suggesting potential avenues for the SVM-ANN hybrid model to incorporate behavior-based elements. (Ahsan, et al., 2022) critical examination of machine learning models' limitations and (Rana, et al., 2022) specialized intrusion detection for zero-day attacks provides a backdrop for assessing the SVM-ANN hybrid model's efficacy. Finally, (Kaushik, et al., 2022) stress on preprocessing for network intrusion detection in the cloud emphasizes the significance of preprocessing steps, offering insights for potential enhancements in the SVM-ANN hybrid model's accuracy and efficiency.

In conclusion, the findings from the experiments integrate into the broader literature, contributing valuable benchmarks and insights for the continuous improvement of intrusion detection methodologies in cloud security.

7 Conclusion and Future Work

In conclusion, the SVM-ANN hybrid model has proven to be a robust and effective solution for intrusion detection in cyber security. The model showcased commendable performance across diverse datasets, achieving high accuracy, precision, recall, and a well-balanced F1 score. The consistency of metrics and alignment with existing literature underline the reliability of the SVM-ANN hybrid model in various scenarios. While acknowledging variations in execution times and the trade-off between false negative rate and detection rate, the research provides valuable insights into the model's strengths and areas for refinement. The findings contribute significantly to the field of intrusion detection, offering benchmarks for practical applications and paving the way for continuous improvement in cyber security methodologies.

The future work should focus on refining and optimizing the SVM-ANN hybrid model to address specific challenges identified in this research. Firstly, efforts should be directed towards optimizing the model for real-time applications through the exploration of parallel processing or advanced optimization algorithms. Secondly, the trade-off between false negative rate and detection rate warrants a more nuanced approach, involving fine-tuning of

model parameters and domain-specific adjustments. Thirdly, the integration of additional models, inspired by successful approaches in related studies, could further enhance accuracy and adaptability.

Additionally, incorporating behavior-based elements and refining preprocessing techniques based on insights from network intrusion detection studies could contribute to improved efficiency. Addressing emerging security challenges, such as zero-day attacks, and exploring hybrid approaches tailored to specific intrusion scenarios represent meaningful avenues for future research. Overall, this research lays the groundwork for continued advancements in intrusion detection methodologies for cyber security, emphasizing the need for a holistic and adaptive approach in the face of evolving cyber threats.

References

Abdelsalam, M., Gupta, M. & Kimmell, J. C., 2021. Analyzing machine learning approaches for online malware detection in cloud. *2021 IEEE International Conference on Smart Computing (SMARTCOMP).*

Ahsan, M. et al., 2022. Cybersecurity threats and their mitigation approaches using Machine Learning—A Review. *Journal of Cybersecurity and Privacy,* 2(3), pp. 527-555.

Aldallal, A. & Alisa, F., 2021. Effective intrusion detection system to secure data in cloud using machine learning. *Symmetry,* 13(12).

Alzahrani, A. O. & Alenazi, M. J. F., 2021. Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks. *Future Internet,* 13(5), p. 111.

Asif, M. et al., 2022. MapReduce based intelligent model for intrusion detection using machine learning technique. *Journal of King Saud University - Computer and Information Sciences,* Volume 34, pp. 9723-9731.

Aslan, O., Gupta, D. & Ozkan-Okay, M., 2021. Intelligent behavior-based malware detection system on cloud computing environment. *IEEE Access,* Volume 9, pp. 83252-83271.

Bera, P. et al., 2020. Machine learning based malware detection in cloud environment using clustering approach. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT).*

Chen, S., Dongxi, L., Luo, S. & Shaukat, K., 2020. Cyber threat detection using Machine Learning Techniques: A performance evaluation perspective. *2020 International Conference on Cyber Warfare and Security (ICCWS).*

Delplace, A., Hermoso, S. & Anandita, K., 2020. Cyber Attack Detection thanks to Machine Learning Algorithms. *arxiv.*

Du, X. et al., 2020. A distributed deep learning system for web attack detection on Edge Devices. *IEEE Transactions on Industrial Informatics,* 16(3), pp. 1963-1971.

Gander, M. et al., 2013. Anomaly Detection in the Cloud: Detecting Security Incidents via Machine Learning. *Trustworthy Eternal Systems via Evolving Software, Data and Knowledge,* pp. 103-116.

Kaushik, C., Ritvik, C., Ram, T. & Lakshman, T., 2022. Network security with network intrusion detection system using machine learning deployed in a cloud infrastructure. *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC).*

Kumar, A. et al., 2022. An intrusion identification and prevention for cloud computing: From the perspective of deep learning. *Optik,* Volume 270, pp. 170044-170055.

Li, K. et al., 2013. Assessment of Machine Learning Algorithms in Cloud Computing Frameworks. *IEEE Systems and Information Engineering Design Symposium,* pp. 90-103.

Lim, S. Y., Kiah, M. M. & Ang, T. F., 2017. Security Issues and Future Challenges of Cloud Service Authentication. *Acta Polytechnica Hungarica,* 14(2).

Meryem, A. & Ouahidi, B. E., 2020. Hybrid intrusion detection system using machine learning. *Network Security,* 2020(5), pp. 8-19.

Ou, C.-M., 2019. Host-based Intrusion Detection Systems inspired by Machine Learning of Agent-based Artificial Immune Systems. *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA).*

Rabbani, M. et al., 2020. A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing. *Journal of Network and Computer Applications,* Volume 151, pp. 102507-102519.

Rana, P. et al., 2022. Intrusion Detection Systems in cloud computing paradigm: Analysis and overview. *Complexity,* pp. 1-14.

SaiSindhuTheja, R. & Shyam, G. K., 2021. An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. *Applied Soft Computing Journal,* Volume 100, pp. 106997-107008.

Saranya, T. et al., 2020. Performance analysis of machine learning algorithms in Intrusion detection system: A review. *Procedia Computer Science,* 171(133), pp. 1251-1260.

Sethi, K., Kumar, R., Prajapati, N. & Bera, P., 2020. Deep Reinforcement Learning based Intrusion Detection System for Cloud Infrastructure. *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS).*

Singh, G. & Khare, N., 2021. A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. *International Journal of Computers and Applications,* 44(7), pp. 659-669.

Soni, D. & Kumar, N., 2022. Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy. *Journal of Network and Computer Applications,* Volume 205, pp. 103419-103458.

Thilagam, T. & Aruna, R., 2021. Intrusion detection for network based cloud computing by custom RC-NN and optimization. *ICT Express,* Volume 7, pp. 512-520.