National College of Ireland

Research Report

Adaptive Machine Learning for Real-Time Intrusion Detection in Networks

MSc in Cybersecurity

Anju Sobha
Student ID: x23298855

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Anju Sobha
Student ID:	x23298855
Programme:	MSc in Cybersecurity **Year:** September 2024
Module:	MSc (Research) Practicum
Supervisor:	Michael Prior
Submission Due Date:	11th August 2025
Project Title:	Adaptive Machine Learning for Real-Time Intrusion Detection in Networks
Word Count:	9069 **Page Count:** 26

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:	Anju S
Date:	11th August 2025

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):	

Adaptive Machine Learning for Real-Time Intrusion Detection in Networks

Abstract

With the complexity of the cyber threats, the issue of developing more intelligent and flexible systems that can detect the known and unknown intrusions in real-time has become a burning matter. The paper suggests a hybrid threat detection scheme that combines the different data sampling techniques and imbalanced data treatment with the classical machine learning (ML) models in such a way that IDS performance becomes more efficient and precise to make a generic space optimised and self-learning cyber security module. Classic ML models like a Random Forest, support Vegetable Machines (SVM), and Neural Network have been extensively applied in the intrusion detection because they have the capacity to learn a pattern based on the past network traffic data. Nevertheless, they are constrained in scaling to high-dimensional data, identifying high-dimensional data or in recognizing zero-day attacks. This work implements test cybersecurity data, such as NSL-KDD etc., to gain an insight into the performance comparison of classical and quantum-enhanced models. Normalization and dimensionality reduction techniques of data pre-processing are used to provide the best possible features representation of classical and quantum classifiers. The findings reveal that classical models are performing well with the traditional forms of attacks, to identify the slightest anomalies and attack vectors complicated in nature with better accuracy and reduced false-positive rates in predetermined cases.

Keywords: Classical Machine Learning models, Data Imbalance techniques, Feature extraction, Threat Detection, Hybrid Model, Anomaly Detection, Network Security

Chapter 1: Introduction

Since digital environment remains dynamic, the sophistication and rate of cyber threats to critical systems and data keeps advancing as well. Whether stratified government systems or enterprise networks, the gravity of today attack, including zero-day exploits and advanced persistent threats (APTs) as well as polymorphic malware, poses a great threat to traditional security systems. In this regard, Intrusion Detection Systems (IDS) form an important front in the collaboration of networks in which unauthorized or malignant activities can be detected. Nevertheless, the common legacy IDS are supposed to be based on the use of predetermined rules or indicator dependent detection techniques, which are short on dynamically identifying brand new or changing malevolent activities (**Abreu, D., Rothenberg, C.E. and Abelém, A., 2024, June**). With these weaknesses, the use of machine learning (ML) approach is rapidly on the rise to address them. The use of classical ML algorithms like Support Vector Machines (SVM), Random Forests and Deep Neural Networks in IDS is growing to learn previously encountered situations and identify anomalies. Those models offer better flexibility and automation, and yet they do not perform well with large-dimensional data, large amounts of traffic and low-profile attack patterns. Also, classical systems may have high false positive and a low efficiency to identify unknown threats. Another approach to the existence of a classical and quantum machine learning approach will be discussed in this paper, a hybrid IDS framework integrating the advantages of both. The probability of guaranteeing accuracy, scalability, and flexibility of the proposed system in the detection of various cyber threats is through the guided usage of benchmark datasets, including NSL-KDD or CICIDS2017 (**Nguyen, M.T.A., Tong, V., Souihi, S.B. and Souihi, S., 2025**).

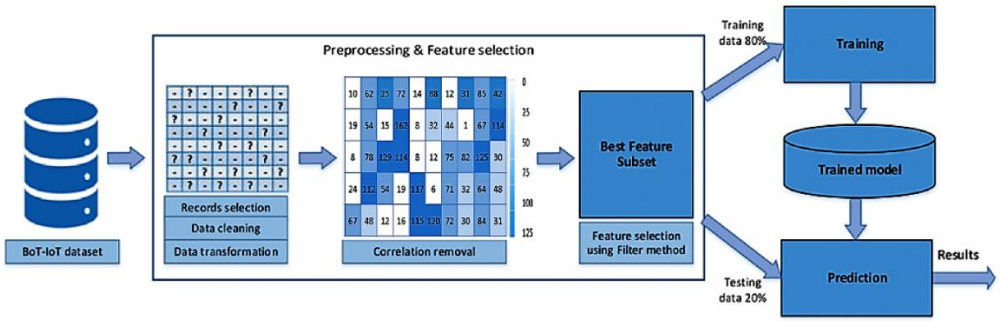


Figure 1: A detailed framework for the real time anomaly detection (**Source: MDPI**)

1.1 Aim of the Project

The project aims at overcoming the drawbacks of the conventional IDS as high false detection rates, low flexibility, and ineffectiveness in identifying unknown or sophisticated attacks through the potentials of classical computing. Structured network data will be analysed using classical ML models, such as Support Vector Machines, Random Forests, and Neural Networks, whereas unstructured high-dimensional and complex datasets will be analysed with the potential to process such data with greater level of efficiency in terms of computation. Such benchmark datasets should include NSL-KDD, which will be training and evaluated. Performance assessment of the hybrid system will be carried out in terms of accuracy, precision, recall and false-positive rate (**Mondragon, J.C., et. al., 2025**)(**Kunang, Y.N., et. al., 2024**)(**Al-Shehari, T., et. al., 2024**).

1.2 Objective

This project aims to create a strong and intelligent Intrusion Detection System (IDS) that will be able to effectively introduce the classical machine learning methods to enhance the detection and classification of cyber threats, especially in a complex network environment (**Mondragon, J.C., et. al., 2025**). The project aims to prove that classical hybrid architecture can substantially increase responsiveness, scalability, and intelligence of IDS solutions and thus, they will be more ready to respond to cybersecurity threats in the present and in the future (**Hdaib, M., Rajasegarar, S. and Pan, L., 2024**).

1.3 Motivation

This project is inspired by the fact that there is already an urgent and increasing demand to improve cybersecurity systems to withstand the mounting and globally spread cyber-attacks due to their frequent, large scale and advance characteristics (**Hdaib, M., Rajasegarar, S. and Pan, L., 2024**). Although traditional Intrusion Detection Systems (IDS) are necessary when it comes to monitoring and protecting the network environment, such systems are mostly rule-based or signature-based and cannot detect new or zero-day attacks very effectively. Despite the fact that classic machine learning approaches, such as Support Vector Machines, Decision Trees, and Neural Networks, have brought intelligent and adaptive components into IDS models, they as well encounter the tremendous challenges that involve massive and high dimensional data as well as finer-grained attack behaviours. In addition, false-positive rates are high, efficiency of the system in real time is slow, which has remained a setback when using such systems. The project hopes to make significant contribution in development of resilient, adaptive and intelligent cyber defence mechanisms by combining the strengths of traditional algorithms, which have been tested over time, and quantum technology which holds promise (**Inayat, U., Ayesha, R. and Mahmood, S., 2024**).

1.4 Problem Statement

The main issue that this project tries to solve is that the existing Intrusion Detection Systems (IDS) fail to offer a possible solution to the emergence of more complex and varied cyber threats to respond to

them in real-time. Despite the goodness of the application of classical machine ML in enhancing the adaptability and intelligence of the IDS, these models undergo fatal limitations. They regularly suffer high dimensional and large-scale network traffic data leading to a higher workload, a slow reaction time, as well as to a decrease in detection precision (**Cultice, T., Onim, M.S.H., Giani, A. and Thapliyal, H., 2024)**. Moreover, classical ML models are inclined to give a high falsely positive outcome, and they are not originally prepared to address the constantly changing threat landscape. The issue then is on the absence of a scalable, precise and adaptable IDS frame that can utilize the classical as well as quantum computational capabilities to identify threats more intelligently and efficiently. The proposed project will fill this gap and advance hybrid IDS based on combining classical ML with the objective to promote a significant increase in the quality of threat detection, false positives, and a path to quantum-secure cybersecurity infrastructure (**Al-Shehari, T., et. al., 2024**)(**Nemati, Z., Mohammadi, A., Bayat, A. and Mirzaei, A., 2024**).

1.5 Research Question

Since we are looking into the world of the detection of, the anomaly using the classical machine learning techniques and focusing upon the two primary things,

(a) How to optimise the space and
(b) How to detect in the real time scenarios without overloading the RAM usage so that the system doesn't become overloaded?

Following is the research question,

RQ: Can classical machine learning techniques, enhanced with feedback mechanisms, effectively improve intrusion detection in Anomalies with enhanced feature engineering techniques for anomaly detection?

In this project we will be making a progress towards a generic framework for the anomaly detection taking into account different features extraction techniques, machine learning models and the data imbalance techniques.

Chapter 2: Literature Review

Threat hunting is an asset in the changing cybersecurity environment as an appropriate measure to safeguard the systems against advanced cyber-attacks. Although conventional security mechanisms should not be neglected, they have become limited to respond adequately to malicious activities and their more sophisticated techniques.

2.1 Approaches and analysis done towards anomaly detection in a hybrid way

Smart City (**Heese, R., Gerlach, T., Mücke, S., Müller, S., Jakobs, M. and Piatkowski, N., 2025**) security must focus on data privacy, security issues in IoT systems, possible cyber risks, and potential dangers to urban assets, community members, and important services, using solid solutions. Examples of solutions are using powerful cybersecurity tools, encrypting important data, deploying AI to find risks, collaborating with others, implementing clear rules, and engaging everyone for a safe and strong smart city environment. For example, SIEM technology allows businesses to check for threats, detect them right away, and act on them because it brings all their security data together.

New kinds of anomalies are always appearing in today's fast-changing world, so classical machine learning cannot stop every threat. ML is becoming popular for its ability to spot problems more efficiently than other tools. In our work, (**Santa Barletta, V., Caivano, D., De Vincentiis, M., Pal, A. and Scalera, M., 2024**) have discussed QML (Quantum Machine Learning) and its role in catching

anomalies in consumer electronics. They have presented a common method to use these algorithms in identifying anomalies.

All of the frameworks are likely to detect network traffic anomalies well (**Hdaib, M., Rajasegarar, S. and Pan, L., 2024**), though the framework using an autoencoder and a k-nearest neighbor is the strongest. Simply put, this shows that quantum approaches could be a hopeful addition for identifying threats in networks, making them more important for upcoming security technologies. Two novel quantum autoencoder frameworks for anomaly detection are discussed below.

2.2 Research Gaps and Comparative Insight Between Classical ML and QML for Anomaly Detection

Later after doing various research identified that despite the rapid advancements in Quantum Machine Learning (QML) (**Cultice, T., Onim, M.S.H., Giani, A. and Thapliyal, H., 2024**) for cybersecurity and anomaly detection, several research gaps persist. First, scalability and hardware limitations significantly hinder QML deployment in real-world networks, as current quantum systems lack the fault-tolerance and qubit stability required for production-level inference. Moreover, data encoding remains a substantial bottleneck; transforming high-dimensional classical data into quantum states without loss of information or increased computational burden is still inefficient. There is also a lack of standard benchmarking, making it difficult to uniformly evaluate QML models across various datasets and application domains. Many proposed frameworks rely on hybrid quantum-classical systems, blurring the isolated benefits of QML over traditional models.

In contrast, classical machine learning (ML) (**Al-Shehari, T., et. al., 2024**) models still outperform QML in multiple practical scenarios. Classical ML frameworks are more mature, scalable, and computationally efficient, particularly when handling large, noisy, or imbalanced datasets, as seen in techniques like SMOTE, ensemble learning, and deep learning with transfer learning. Classical models also benefit from a rich ecosystem of explainability tools (e.g., LIME, SHAP) and a well-established suite of benchmarking datasets and evaluation metrics, which enhances trust and interpretability.

2.3 Analysis of feature extraction techniques in IDS

In particular, (**Balla, A., Habaebi, M.H., et, al, 2023**) incorporates CNNs with three of the most common oversampling algorithms to data imbalances: ADASYN, SMOTE, and Borderline-SMOTE. The research question will be to better understand the detection accuracy and resilience to the insider attacks with the use of an imbalanced data such as those used in cybersecurity. As opposed to SMOTE and Borderline-SMOTE, which work with minimal data which are considered to be imbalanced datasets in order to increase the effectiveness level of detection, ADASYN with the help of CNN achieves the ROC curve proportions of nearly 96% as was demonstrated in the experiment. The results of these three hybrid models that solve imbalance in CNN are compared to the current best works with respect to accuracy rate, recall, and ROC. The results of our study ought to be included in the insider threat detection procedure.

In the security domain (**Lopez-Ledezma, M. and Velarde, G., 2024**), some of the important monitoring applications are intrusion detection systems (IDSs) and intrusion prevention systems (IPSs). The problem of data imbalance remains even despite countless works that have been conducted to address it by employing the usage of deep learning methods to build an effective Intrusion Detection System (IDS). The research is aimed at investigating how data imbalance will impact the effective design of a SCADA based intrusion detection system. The Morris power and the CICIDS2017 datasets, due to the application of a random sampling process, a one-sided selection (OSS) process, near-miss process, SMOTE, and ADASYN process were identified as two imbalanced datasets used to study the impacts of different data balancing methods. Long short-term memory (LSTM) and convolutional neural networks (CNNs) were used to classify into binary using a combination of the models. The confusion

matrix contains such metrics of evaluation as F1-score, accuracy and precision, and detection rate used to analyze the viability of the system. In the 4 experiments, using two sets of data, there was surely a problem of data imbalance. The research is going to help future investigation examine how imbalanced data reflect on Deep Learning-based SCADA Intrusion Detection Systems.

In Credit Card dataset (**Nemati, Z., Mohammadi, A., Bayat, A. and Mirzaei, A., 2024**), 0.2 percent, out of a total of 283726 samples, are fraudulent transactions and as such the ratio of imbalance in this dataset is 598.84;1. The records are 31 attributes. In PaySim dataset, there is a ratio of 773.70:1 imbalance ratio in which there are 0.13 percent of all the samples, i.e. 6,362,620, that are fraudulent transactions. There are 11 attributes in the dataset. This paper presents three experiments. The first experiment examines some well-known methods, such as Random Forests, eXtreme Gradient Boosting, Decision Tree, Gradient Boosting Decision Tree, Light Gradient Boosting Machine, and Logistic Regression.

Initial performance (**Mahboubi, A., et. al., 2024**) of the SVM classifier on all financial ratios is not to our expectation. This therefore gives us and all rounded approach to anomaly detection and the versatility and effectiveness of our methods in the analysis of financial data streams and network security. Specifically, GWO is remarkable, in terms of fitness function, 0.2940 and accuracy of 70.06 percent after 31 iterations. These 9 very important financial ratios are extracted successfully by this algorithm. These ratios extracted by GWO are made to be incorporated into the SVM classifier, and the anomaly detection model obtains excellent accuracy, precision, reduction of errors, and performance with 75.83 percent, 66.80 percent, 33.2, and 80.3 percent respectively.

In this paper (**Dong, J.,et. al., 2025**), the importance of threat hunting as a step-by-step process, conducted by an analyst to identify invisible threats that transform an organizations digital infrastructure into a significant incident, is discussed. Although the cybersecurity positively affects security, the community struggles with several issues, such as the absence of identical approaches, specific knowledge, and the introduction of advanced technologies, such as artificial intelligence (AI) to practice predictive threat detection.

2.4 Enhanced end to end pipeline for the anomaly detection in the real time

In addition, (**Lee, K.S., Kim, S.B. and Kim, H.W., 2023**) should address the model-based service implementation to the stakeholders during the data collection in real-time and also the deployment of a model. A deep learning-based model of anomaly detection, developed by our research, is a hybrid model which does not require data labeling. The strength of this model has been in its ability to detect odd patterns through the fact that it can reconstruct the sequence flow in the given data. After conducting the experiment, the suggested hybrid model was proven to be more effective than the other types of anomaly detection algorithms as what our experimental results depicted. In knowing in advance when the manufacturing process may experience a hitch, it helps to increase the efficiency of the production process. In addition, we create an anomaly detection system using a real-time service framework of the data collection step to enhance the activation of smart factories.

The majority (**Karim, H., Doshi, K. and Yilmaz, Y., 2024**) of those approaches involve ad hoc feature aggregation rules and utilize metric learning losses whose weakness lies in the fact that they restrict the models to notice anomalies in real time. As per the assumption of deep neural networks, an emerging interest has also been in constructing end-to-end methods which can automatically learn to extract good features directly out of the raw data. This is contrary to what is currently there that makes use of learnt feature extractors largely. The described approach is much faster and has better AUC results than the known methods in anomaly detection. In particular, on the UCF-Crime dataset, our method can obtain 86.94% AUC with the decision period of 6.4 seconds whereas the competing methods can yield at most 85.92% AUC with a decision period of 273 seconds.

The structure (**Nazat, S., Li, L. and Abdallah, M., 2024**) provides explanations of the black-box AI models globally and locally with the help of two XAI techniques. Further, two new feature selection methods are proposed to highlight the important features used to detect anomalies based on the prominent SHAP XAI approach and prediction accuracy of various black-box AI models. We benchmark our proposed feature selection methods against six state-of-the-art feature selection methods (that include two wrapper based feature selection methods) and show the higher score on several measures of the evaluation processes. In order to get an overall effect of our feature selection schemes, we use three separate classifiers in assessing our proposed feature selection schemes. The new methods of feature selection are useful in extracting the most informative features improving the explainability of the models.

The observation (**Rezaee, K., 2024**) on security and protection of the public places is the critical part in the problem that the crowded areas could not be controlled manually because they are unpredictable and involve complex issues. The algorithms out of the abnormal behavior have tried to enhance efficiency, defend against pixel occlusion, generality, calculating complexity, and time of execution. Comparable to the state-of-the-art of abnormal behavior detection of crowd videos, we generally categorized into several groups like tracking, classification based on the hand-crafted extracted features, classification based on the use of deep networks and hybrid models. The classification stage has been identified to produce foe satisfying outcomes in hybrid and deep learning. In this study, a pool of video frames referred to as Motion Emotion Dataset (MED), is used to analyse the different conditions that govern these methods. The analysis of crowd and individuals behavior to screen abnormal events can be achieved by incorporation of an appropriate real-time approach and consideration of WoT platform.

2.5 Research Gaps

Although both traditional machine learning and the emerging quantum techniques have already made a breakthrough within the domain of anomaly detection, a number of issues are still to be addressed. A consistent approach to test different hybrid and QML-based intrusion detection systems over different datasets has been a major issue and must be taken care of. Regarding the issues of real-time applicability in real-time quantum computing, it is not without oneself needed to get rid of such aspects as data encoding, scalability, and limitation of hardware. Moreover, when dealing with complicated and dynamically changing environments like smart cities or significant infrastructure, interpretability and actual applicability is normally overlooked in previous research. This is generally so. Finally, most of the models that are currently implemented today are tested under controlled conditions, instead of adversarial or zero-day attack conditions. Due to this, they cannot generalize and excel in situations that rely on the real world.

2.6 Research Outcomes

This paper will address the issue of increasing the detection of vulnerabilities and anomalies within the NSL-KDD data set, through the use of traditional machine learning strategies. The proposed method is a less complex and explainable intrusion detection system to ensure the higher precision of identification and the reduced false-positive rate. It is supposed to be deployed in a real-time environment. Unlike deep learning, classical models are simpler and require less computation and provide more transparency. The salient aspects are the complete assessment of algorithms, the emphasis on balanced optimization methods like SMOTE aimed at curbing during class imbalance, and the integrated Resilience and anomaly detection. This paper gives scalable model of proactive threat hunting in cyber security which bridges the gap between academic works and industry practice.

Chapter 3: Methodology

Network security has become a much sought-after corporate concern nowadays with the fortifications against networked systems becoming a priority. The whole concept behind this is the ever-increasing

links between networked systems. Keeping computer networks safe and ensuring their integrity has become paramount, with cyberattacks ranging from petty theft of information to gross invasions by malware. They always have varied methods in which they go about breaching computer networks. Against contemporary instances concerning network security, attacks have a conventional approach and have weaker defenses for nonconventional methods that rapidly arise. Even when these methods work, they sometimes remain ineffective. Now, any past slight in approach could be an opportunity for an attacker to compromise network security.

3.1 Dataset Description

The NSL-KDD dataset uses a more superior training set in comparison to the original KDD dataset as the training set contains no duplicate entries which in turn mean that the classifiers will no longer give predilection to the most common instance. To ensure that there is no biasness in the performance of learners as to which algorithms would have a higher detection rate on the prevalent records, the proposed test sets possess adequate records, and they are not duplicated. https://www.unb.ca/cic/datasets/nsl.html. The percentage of every level of difficulty in the original KDD dataset is negatively correlated to the numbers of records chosen out of that level. The differing categorization rates of diverse machine learning algorithms make it possible to make accurate comparisons of learning methodologies. Since training and testing sets are of massive size, there is no need to pick a tiny group and use it to test the whole data set. This way, many studies will get similar and vicinal conclusions of assessment.

Dataset	Number of Records:					
	Total	Normal	DoS	Probe	U2R	R2L
KDDTrain+20%	25192	13449 (53%)	9234 (37%)	2289 (9.16%)	11 (0.04%)	209 (0.8%)
KDDTrain+	125973	67343 (53%)	45927 (37%)	11656 (9.11%)	52 (0.04%)	995 (0.85%)
KDDTest+	22544	9711 (43%)	7458 (33%)	2421 (11%)	200 (0.9%)	2654 (12.1%)

Figure 2: Data types in KDD dataset (**Source: medium**)

The detailed information of the dataset present in the dataset is as below,

Dataset Link: https://www.unb.ca/cic/datasets/nsl.html

3.2 Machine Learning Algorithms

The project seeks to enable users to actively recognize and address security risks by training AI using datasets like NSL-KDD for intrusion detection and CVE for vulnerability assessment. The project seeks to validate the solutions developed to improve system security and resilience against cyber threats by applying an AI-assisted simulation testing framework in the operational environment and establishing monitoring and response systems.

3.2.1 Logistic Regression – Linear Based model

Our prediction research on simulation testing uses Logistic Regression (LR)(**Nick, T.G. and Campbell, K.M., 2007**), a nice machine learning method that is uncomplicated and easy to comprehend. LR predicts probabilities of a binary outcome, such as Suspect or Correct, based on input features. In terms of understanding important factors that determine simulation testing with the AI -- based options (e.g., Correct, Suspected) LR is not only a justified method but allows one to understand the relationship between the predictors and the dependent variable. Despite being relatively simple, LR is a strong baseline approach because it can be effective and efficient for binary classification problems.

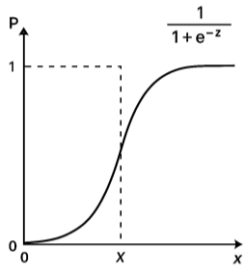


Figure 3: The S-Curve analysis for the Logistic regression.

The logit function is as below,

$$f(z) = \frac{1}{1 + e^{-z}}$$

When this above is differentiated, then

$$f'(z) = f(z)(1 - f(z))$$

Making the calculations easier. Here *f(z)* is the logit function.

3.2.2 Decision Trees

In our research, we employ decision tree (DT) (**De Ville, B., 2013**) algorithms due to their interpretability and ability to manage non-linear interactions. DTs create a decision tree-like model by splitting data into subsets based on the most relevant features. This is useful for understanding decision rules in simulation testing using the AI approval approaches, as it is very easy to visualize and understand. However, pruning techniques and ensemble methods can avoid DT's overfitting.

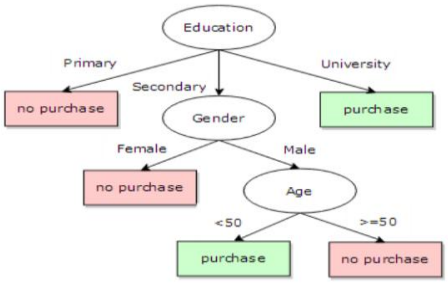


Figure 4: Example of a decision tree. The decision tree has three internal nodes. At each internal node a decision is made based on an attribute value. To classify a new instance, it is evaluated down the tree, making a decision at each internal node and resulting in one of the leaf nodes. (**Kuznetsova, N., Westenberg, M., Buchin, K., Dinkla, K. and Van Den Elzen,S.J., 2014**)

3.2.3 Random Forest

Random Forest (RF)(**Rigatti, S.J., 2017**) is an ensemble model method that creates numerous trees during the training phase, and the default output is the mode of the classes for classification problems; which enhances decision tree predictive performance. RF diminishes the chance of overfitting by averaging the output of multiple trees developing a robust, accurate model. In addition, RF adequately manages a high number of input features, allows for good ability accuracy, and is therefore well suited to our simulation test utilizing the AI prediction task.

Figure 5: Random Forest with 100 trees(**Kuznetsova, N., Westenberg, M., Buchin, K., Dinkla, K. and Van Den Elzen, S.J., 2014.)**

3.2.4 Support Vector Machines

We use Support Vector Machine (SVM) (**Vishwanathan, S.V.M. and Murty, M.N., 2002**) in our research due to its robustness in classification tasks and ability to work with high-dimensional data. SVM focuses finding the best hyperplane that maximizes the margin between the classes. "In particular, SVM leads to good generalization performance and high accuracy with respect to a project like ours because particular kernel functions restrict the amount of overfitting that can occur given a certain training set and score as they focus on the process of finding non-linear relationships".

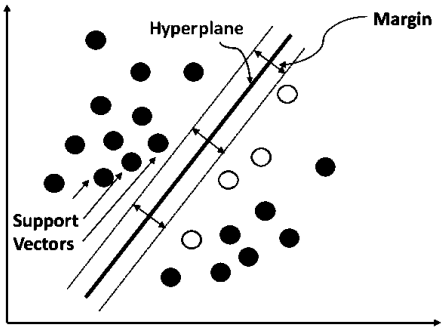
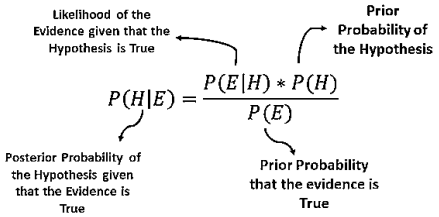


Figure 6: support vectors visualization (**source: researchgate**)

3.2.5 Naïve Bayes

We choose Naive Bayes (NB) (**Rish, I. . 2001, August**) because it is easy to implement and performs well on large folds of text. Naive Bayes follows Bayes' theorem, where the features of the data, and while also considering a class label; are assumed to be conditionally independent. The acknowledgment that NB works extremely well in a real world, speaks to the blind assumption of independence. It will aid us in our research work, given the performance in binary classification and multiclass classification tasks. Regarding performing our original threat detection and spect sporadic screening's NB is suitable given its strength in identifying and separating patterns and anomalous data in network data.



3.2.6 k-Nearest Neighbors

One additional technique that we utilized in our simulation testing platform is the k-Nearest Neighbors (k-NN) algorithm which is extensively documented in (Peterson, L. E. , 2009) and which is very easy and fast for classification. k-NN estimates the position of a data point based on the positions of the k nearest neighbors in the feature space based on the majority class. The k-NN method is flexible and easy to perform because it does not assume anything about the form of the distribution of the data we are working with.

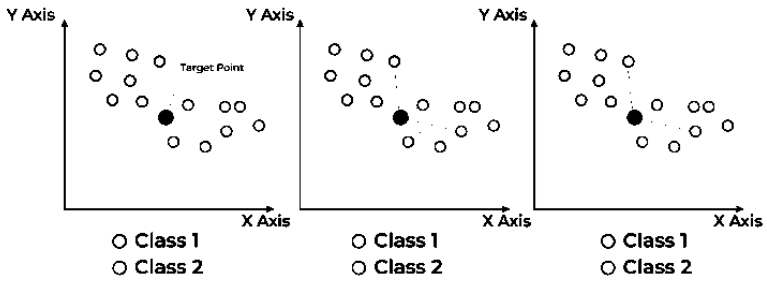


Figure 7: Working of K-NN Algorithm (**Source: Geeksforgeeks**)

3.2.7 LSTM – Long Short-Term Memory

During the simulation testing platform development phase, LSTM networks are used (**Yu, Y. , Si, X. , Hu, C. and Zhang, J., 2019**), a type of recurrent neural network (RNN) used for sequential and time series data. LSTM also works very well for time-series of traffic analysis due to their ability to process long-term memories of data. It is important to incorporate network traffic analysis when monitoring for new emergent pathologies across complex attack patterns. The value of our framework is that it can learn the network behaviors using the LSTM networks which helps in identifying complex attacks that may not have been able to be discerned by the individual data set itself.

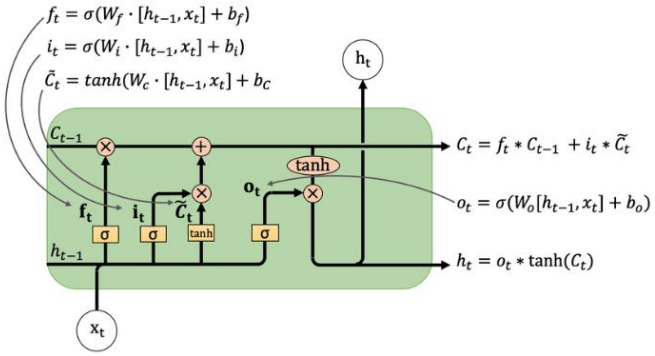


Figure 8: Complete LSTM architecture with equations showing how information moves through the cell (**Source: Researchgate**)

3.2.8 Real Time simulation environment for the anomaly detection and mitigation

Our main goal is to improve the intrusion detection on network in various operating systems by creating an AI-enhanced real-time simulation environment. This install is used to develop and evaluate advanced machine learning algorithms to detect anomalies and find possible vulnerabilities. By modeling realistic attack settings, the system provides not only the better identification of the threat but also beneficial quick mitigation strategies. The methodology, tools, and techniques included in this simulation framework are evaluated in-depth as they are the findings and innovations represented by (Rohit et al., 2023).

3.3 Evaluation Matrix

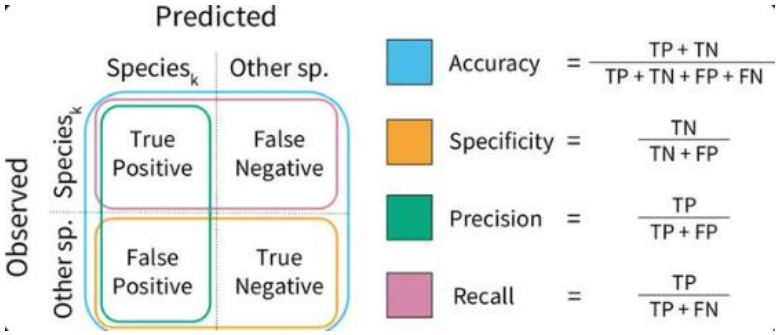


Figure 9: A Quick Overview of Evaluation Metrics for Classification Models (**Source: KDnuggets**)

Chapter 4: Implementation

In this section, we will discuss the steps that we are going to undertake in order to construct the simulation testing system of the project. It includes vulnerability detection with the application of the machine learning (ML) and intrusion detection by using the NSL-KDD dataset. These are its two main features.

4.1 Algorithm Pipeline and the Flow Diagram

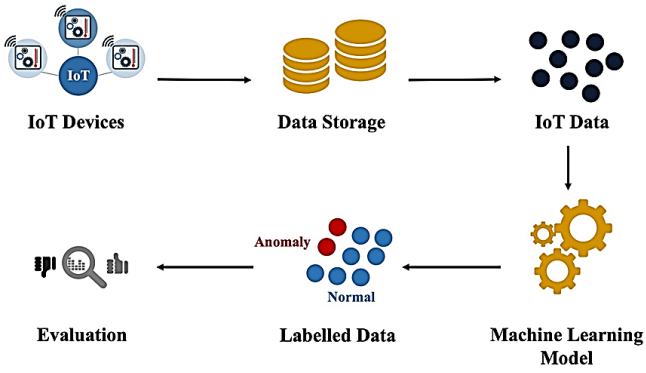


Figure 10: Generic Flow diagram of the pipeline depicting the analysis if the anomaly behaviour and detection of the vulnerability in the real time

To support this the framework uses NSL-KDD dataset and a binary classification algorithm, to perform some analysis on network traffic distinguishing between legitimate traffic and the suspected intrusions. The well-known benchmark datasets of intrusion detection systems, the NSL-KDD includes labeled data which consists of various network traffic patterns classified as normal and different types of attacks. The sequence that follows contains the crucial steps:

4.1.1 Algorithm Description and Steps

The system Name: AI-Powered System that will help to identify and find reducing the anomaly in real time

Input Dataset: NSL-KDD Network Traffic Dataset was used to do this experiment.

Completing Product Output: Revised vulnerability reports and up-to-the-minute threat alerts

4.1.2 Steps Detailed Outline

Step 1: *Sanitize and Transform the Data to Prepare*

(a) We need to clear out irrelevant or redundant items contained in the traffic log, e.g., benign or internal traffic.

(b) Then we use either label encoding or one-hot encoding to turn categorical data into quantitative representation to take advantage of machine learning procedures.

Step 2: *Selection and the Engineering of Features*

(a) We have to identify the most significant traits that can be used to detect anomalies accurately, and use them.

(b) Then we will be keen to reduce the dimensionality so as to improve a model and avoid overfitting.

Step 3: *Training Modeling Data processing*

(a) We gather an all rounded and versatile exercise program.

(b) Divide the data into two groups of a training and validation data.

(c) Make the data pipeline-ready in such a way that a consistent input can be given to the machine learning models.

Step 4: *Binary model of classification*

(a) To segment between the normal and abnormal traffic, you would have to create a binary classification model.

(b) Combine the classifier with a real-time system to monitor the network to do real-time analysis.

Step 5: *Intrusion Detection and Prevention System*

(a) Use the trained classifier to run in a real-time on the traffic generated by the network.

(b) Automated alerts will also be escalated to relevant security personnel in situations where suspicious activities have been identified.

Step 6: *Develop a ML Map and Identify its Vulnerability*

(a) Further tools to check the intrusion alarms include other ML-based vulnerability diagnostic tools to known system threats.

(b) Make up a list of potential areas of weakness that require further research in its order of priority.

Step 7: *Simulation Security Testing*

(a) In regulated conditions, it performed specific test cases and simulated offenses.

(b) Obtain the efficiency of the system and learn how efficiently classifiers perform in various threat situations.

Step 8: *Logs and Threat Reporting Analysis*

(a) Discover the concerning tendencies addressing the analysis of classifier outputs and logs.

(b) Include the latest discoveries in reports together with isochrones, danger ratings, anomalies, and the actions to be taken.

Step 9: *Network containment and incident escalation.*

(a) Introduce firewalls or other isolation solutions within the systems to separate weak systems when a strong projection is at hand.

(b) Instantly bring up a security operations center (SOC) or cybersecurity response team with enterprise deployment.

A Binary Classifier's Use with NSL-KDD Comes with Several Benefits:

• The system continuously monitors network data, which enables quick identification of potential intrusions—this is what we call real-time intrusion detection.

• Leveraging an Existing Database: The NSL-KDD dataset is an excellent option for training the binary classifier.

• Enhanced Accuracy: By employing binary categorization (normal vs. intrusion), the model significantly improves its chances of spotting malicious activities with greater accuracy.

4.2 Stage 1: Identification of Vulnerability presence

Most vulnerability data are stored in the Common Vulnerabilities and Exposures (CVE) repository, which contains the data on identified vulnerabilities with code samples. The information was taken off the Kaggle site. The received data should be pre-processed before carrying out the analysis of the code. The steps used during the preparation of code of the machine learning models included tokenization, removal of comments and conversion of format. Next, code complexity, n-gram, and function calls were obtained as relevant features in the preprocessed code. These properties help the model to identify any possible vulnerabilities.

4.3 Stage 2: Machine Learning based Intrusion Detection and classification of the type of it

This step of integrating the dataset of the NSL data set in KDD dataset that is widely accepted as being among the most notable data sets in the study of intrusion detection systems led to the framework being greatly improved. It can be distinguished between malicious activity and regular network activity due to how this set of information is structured. It was one of the primary functions that it had in the task of performing the feature selection exercise which helped in identifying the features that were most pertinent to the task of intrusion detection. Relevant parameters are protocol being used, the size of a packet, and the Internet Protocol (IP) addresses of the source and destinations that are involved in the transmission and reception of data.

The cleaned up NSL-KDD data set has been used to develop a classification model that has been developed using machine learning. Under this model, the most used type of model is the Random Forest, a Decision tree, or even a Neural Network. In the training program, the model had picked up a trend in the data of its network that would mean that there was an assault risk. The behavior of the target network was also analyzed on a fly using a device which was put up to monitor the traffic on the network. Those were then added into the trained intrusion detection system so that it may continue analysis of them.

4.4 Detailed Data Features outline and Classes understanding

The output attacks are of different types and the numbers of different type counts are listed as below,

Attack Type	Count
normal	67343
neptune	41214
satan	3633
ipsweep	3599
portsweep	2931
smurf	2646
nmap	1493
back	956
teardrop	892
warezclient	890
pod	201
guess_passwd	53
buffer_overflow	30
warezmaster	20
land	18
imap	11
rootkit	10
loadmodule	9
ftp_write	8
multihop	7
phf	4
perl	3
spy	2
dtype:	int64
dtype:	int64

Table 1: Analysis of the attack type

The sphere of cybersecurity has a number of recognized threats and instruments that often interfere in the process of intrusion. A list of distinctive types of attacks and methods of exploiting them, categorized based on trends noted in intrusion detection data sets, is presented below:

- *Neptune Attack*: This is a type of Denial of Service (DoS) attack in which the attacker uses deluge of TCP connection requests to engage a target machine or service such that legitimate users cannot connect to the target.
- *Satan Tool*: The tool is mostly related to vulnerability assessment as it is used to scan networks in order to uncover points of weakness within systems and it is usually adopted by the attackers as part of the reconnaissance phase.
- *IP Sweep (Ipsweep):* A type of network mapping where a range of IP addresses would be scanned to obtain which hosts are active and possibly exploitable.
- *Port Sweep (Portsweep)*: Port scanning technique used an attacker tries to access various ports on multiple hosts in order to identify services that are open, and without protection.
- *Smurf Attack*: This is a variant of the enhanced DoS attack that employs the use of ICMP packets and the address being affected is broadcasted using an address. All responding systems redirect the replies to the victim which uses up its resources.
- *Nmap Scanning*: A network discovery tool that an administrator can use to figure out what is wrong with their network, or an attacker can use to enumerate currently alive hosts, ports, and services.
- *Back door Access (Back):* This is unauthorized remote access that is usually set up by taking advantage of an open vulnerability or feeding in malicious code to circumvent standard authentication.
- *Teardrop Attack*: A DoS legacy exploit method against IP packet reassembly vulnerabilities. Such inconsistency may cause a crash of poorly configured systems.
- *Warez Client*: Refers to a system that deals with distributing software in an inappropriate manner and it is usually in a peer-to-peer pirate sharing of software or files.
- *Ping of Death (Pod)*: One that is similar to resource exhaustion DoS attack, but the novelty is the use of malformed or over-sized packets sent to a system intended to crash or freeze.
- *Password Guessing (Guess_passwd):* An attempt to manually guess the password is a brute-force method by which attackers attempt over and over again to guess the user credentials by using weak or default passwords.

- *Buffer Overflow*: A vulnerability in which memory is corrupted by writing more data into a buffer than it can manage, usually resulting in an opportunity to inject malicious code and execute it.
- *Warez Master*: This term refers to the main source or provider in a warez network and may have the task of managing and seeding pirated material like the client.
- *Land Attack*: It is a relatively new type of DoS attack which utilizes the spoofing technique and consists of sending specially formed packets to the victim machine sharing the same source and destination IP address so that the victim machine becomes stuck in a loop and shuts down.
- *IMAP Exploit*: Attacks the weaknesses of email services based on the Internet Message Access Protocol (IMAP), usually to access email accounts or e-mail servers illegitimately.
- *Rootkit*: An invisible malware toolkit to access gain privileged (root) access anonymously in such a way that the presence is not detected by the tool or viewable by the user.
- *Loadmodule Exploit*: This is the use of injecting malicious modules or malicious codes into a system and running arbitrary commands that are executed in elevated privileges.

- *FTP Write Exploit:* This exploit is utilized to exploit unsecure FTPs with write permission and use them to upload malicious stuff in a server.
- *Multihop*: A technology in an attack, or a means of communicating that involves utilizing more intermediary systems to hide the source of malignant activity or to avoid security policies.
- *Phf Exploit*: Web servers which had previously run old CGI scripts (such as phf) were affected, as it was a command injection exploit that allowed command-level access to the attacker.

Such terms are related to variety of network security threats and system vulnerabilities as usually experienced in networks.

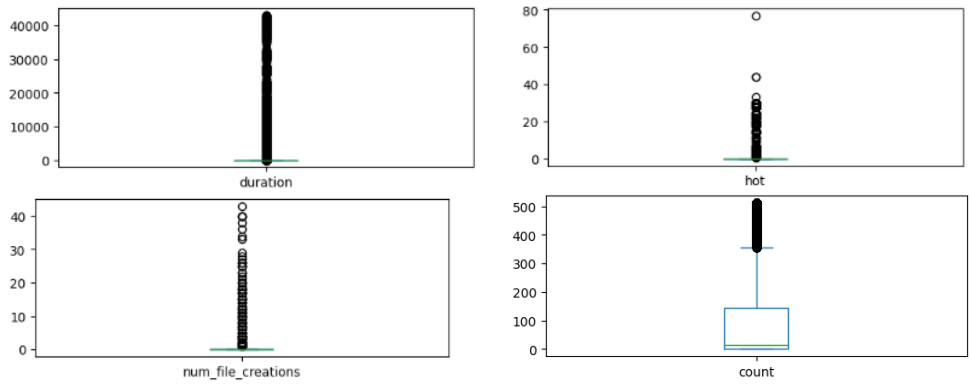


Figure 11: Exploration of the outliers presence in the dataset

The figure shows the points arrangement of the data, and possible anomaly. The statistics representing the data are given by the points coming above the upper whisker of the boxplot which are the statistical outliers that occur in the boxplot and are said to be unusual behavior in the data.

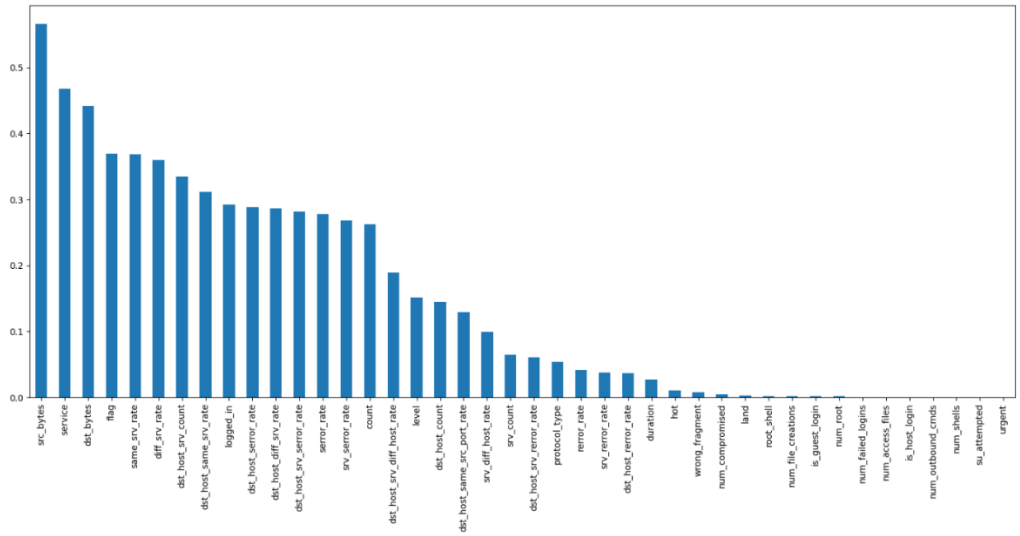


Figure 12: Bar plot of the important features presence for the feature selection.

To reduce the dimensionality and complexity of the system, we employed the feature_importances_ attribute from scikit-learn to identify the most influential features. By focusing on these top-ranked attributes, the model becomes more efficient while retaining critical information for accurate intrusion detection. Top features are taken into the consideration

Chapter 5: Results and analysis

The binary classification model, which was developed in the context of the simulation test which aims at intrusion detection, gained some assessment on the basis of the NSL-KDD dataset. This dataset has a huge variety of network traffic patterns, each of which is either a characteristic of normal traffic or a sign of an attack. The model trained was successful in classification of the records and the results are as follows:

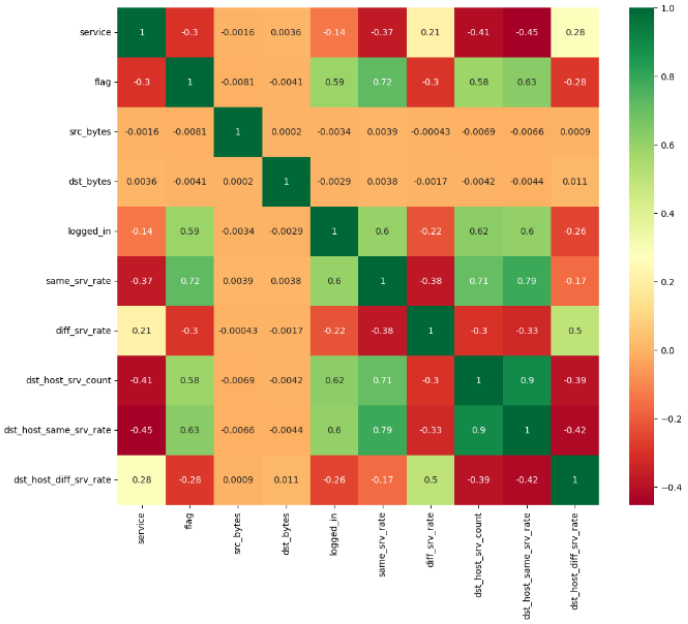


Figure 13: Pearson Correlation Matrix

The diagram above reveals that features have a correlation to one another based on Pearson correlation. This correlation strength is represented in green in those cells. Next is looking at what kind of attack distribution there is in the data:

- Normal Traffic The second segment is normal network traffic that is the most massive part of the dataset with 67,343 entries, taking up about X percent of the complete number of records.
- Attack Categories:
  (a) *Denial-of-Service (DoS) Attacks*: The category of DoS attacks, which is called Neptune, presumably, because of the DoS attack involving flooding, includes 41,214 different devices attacking, which is X percent of all identified attacks.
  (b) *Scanning Attempts*: A group of 3,633 records (X%) is denoted by the name Satan, and it indicates the vulnerability scanning misconducts that are focused on the detection of vulnerabilities.
  (c) *Port Sweeps*: The 3,599 examples (X%) of the class, Ipsweep, imply that they are scouting the open ports on networks via active scanning.

Other types of attacks: The less frequent attacks are; Portsweep, Smurf, Nmap, Back, and Teardrop, which are all minor and don't contribute too much to the idea of potential threats but still are worth the knowledge as well.

5.1 Case 1: Machine Learning-Based Evaluation

The model gives an excellent capacity of the model to differentiate between benign traffic and other malicious actions. The above statistics show the model was able to classify the normal data well because of the high volume of correctly classified normal records (67,343) as compared to the normal data records (71,483). The DoS attacks were the most common found in the detected threats meaning there are attempts to jeopardize network availability. At the same time, reconnaissance-related actions, i.e. those that fall under the category of Satan and Ipsweep, can be interpreted as initial scanning activities before going on with more serious attacks.

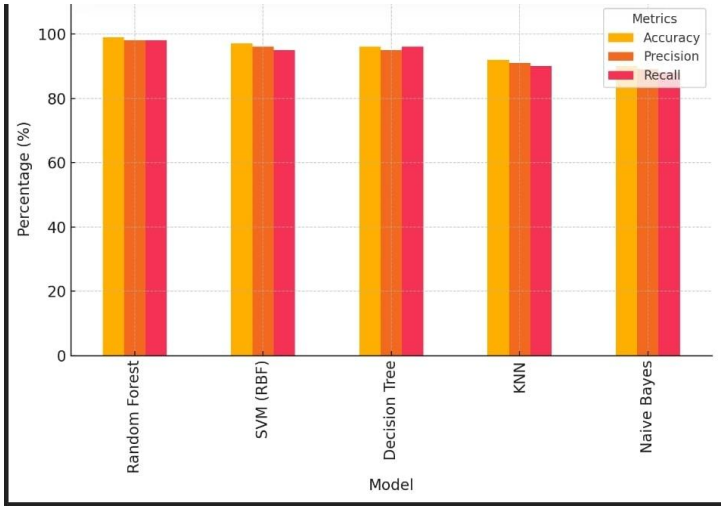


Figure 14: Performance of Machine Learning Algorithms

Random Forest will be the most accurate, precise, and recall model of the tested. Support Vector Machines (RBF Kernel) had remained a competitive approach, in respect to Precision and Recall. K-Nearest Neighbors and Naive Bayes were bad in all the evaluation metrics. Decision Trees are not the leading method in any of the before mentioned parameters, but they did deliver a decent result in all of them.

The ROC analysis also stated that Decision Trees and Random Forests performed best in providing optimal trade-off between the true positive and false positive rates.

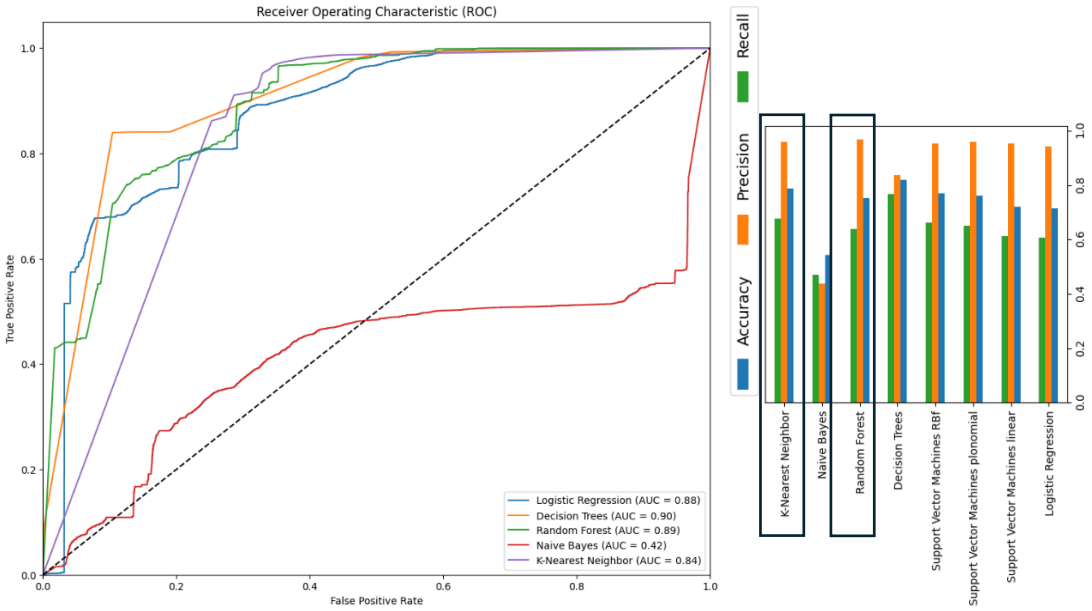


Figure 15: Best model performance curves *a)* ROC Curves for All Models *(b)* Comparative Performance Plots

5.2 Vulnerability detection

The purpose of this analysis was to evaluate whether the different machine learning algorithms can be deployed in relation to the individual software and hardware exploits to identify vulnerable exploit patterns using the CVE data. This is an all-inclusive data held by the MITRE Corporation, regarding the vulnerability data and it includes CVE IDs, platforms, vulnerability types, and possible impacts, and references.

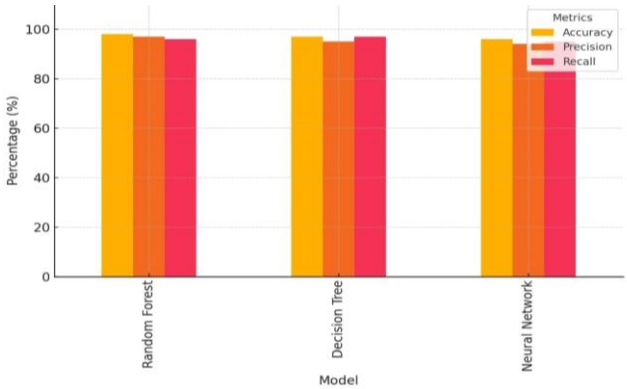


Figure 16: Random Forest, Neural Networks and Decision Trees performance evaluation for the vulnerability type decision making

Random Forest showed the best accuracy and precision as compared to the other models, which implies that the model performs best and flags proper vulnerabilities. The results of Decision Tree were slightly better in terms of Recall, i.e., higher possible coverage of more types of real vulnerabilities. Neural Networks succeeded with average results but perhaps it is possible to adjust some hyperparameters so that the results are improved in Recall and Precision.

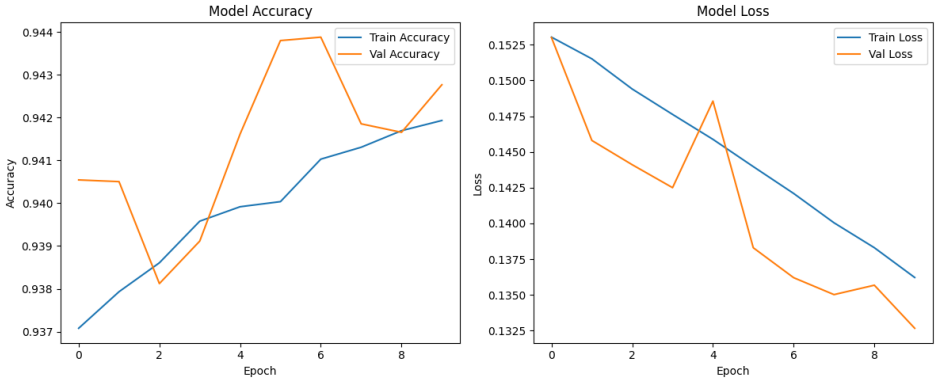


Figure 17: Plotting the loss and accuracy curves to fit the optimisation step

The above graph represents the learning curve of a neural network model after several iterations in the time of training. The model in epoch 10 took about 18 municipalities to record a training loss of 6.58e-5 and a training loss of 3.18e-7 implying that it learned effectively and not overfitting. Nevertheless, despite this kind of outcomes, one has to test model performance on new unobserved data to be certain of real-life effectiveness.

5.3 Attacks mitigation

In this case once a certain type of attack is identified with the help of the trained ML model, a mitigation action is taken with regard to that particular class of attacks. Such a restricted response does not demand such use of irrelevant system resources thus attacks are dealt with in an efficient manner. One of the most important innovations that should be identified with regard to this research is that it saves RAM and processing power by activating only the mitigation required to address the threat that has been detected.

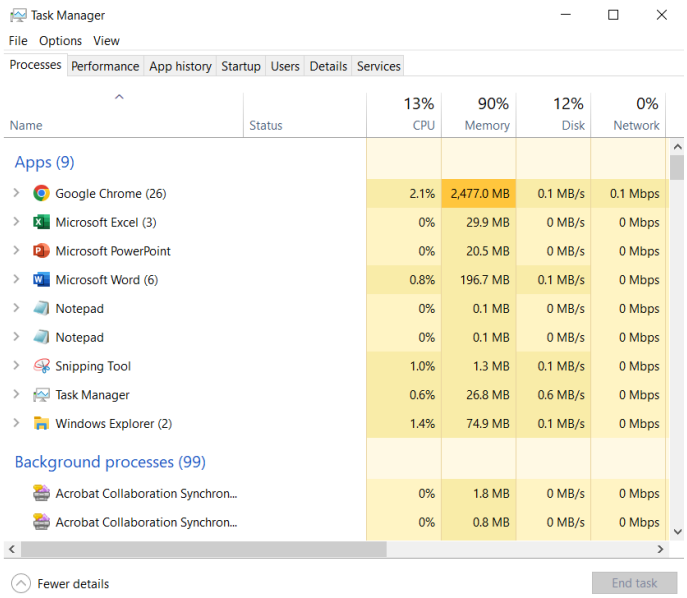


Figure 18: System Memory Usage During Mitigation

We see that our algorithms Has not taken much of the space.

5.4 Real Time Scenario 1: Running without a server, Experimentation using the serverless system

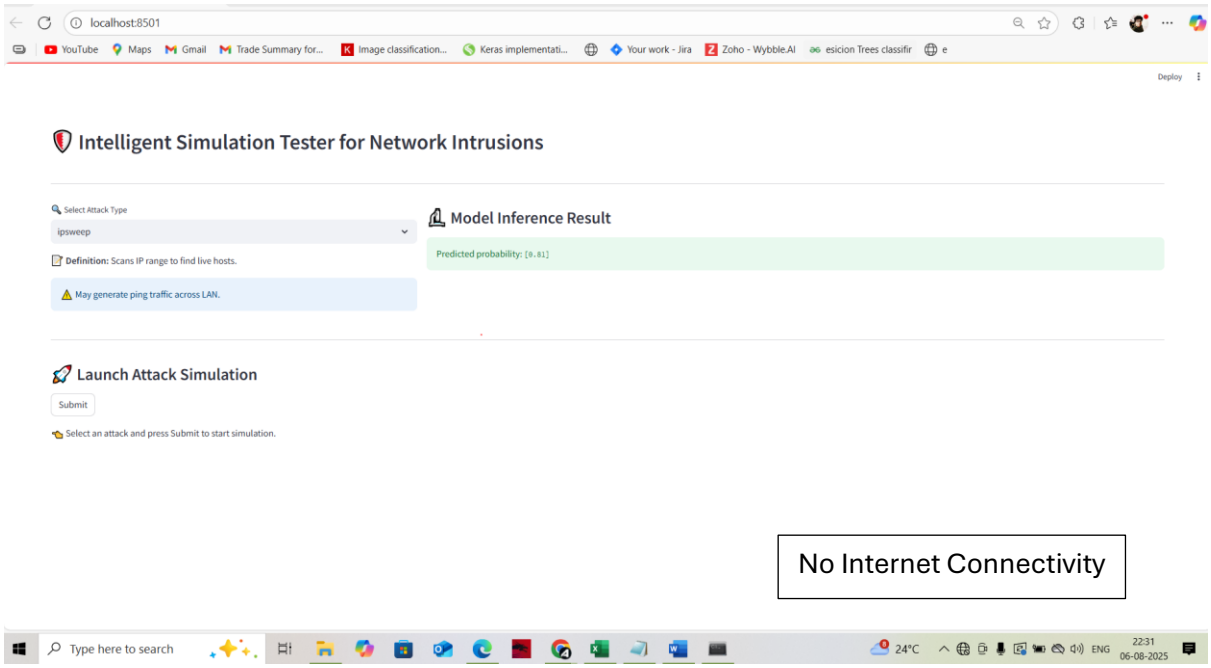


Figure 19: System Operation in Serverless Mode

The intrusion detection system performs effectively without any reliance on internet connectivity or cloud-based services, demonstrating its independence from external networks.

5.5 Real Time Scenario 2: Runing experimentation using different OS

Upon checking on windows 11 pro and windows 10, it was found that the model needs to rely on a dependency on the packages that was imposed when installed, depending on the version of python used. The model worked according to need in the component of performance of the system.

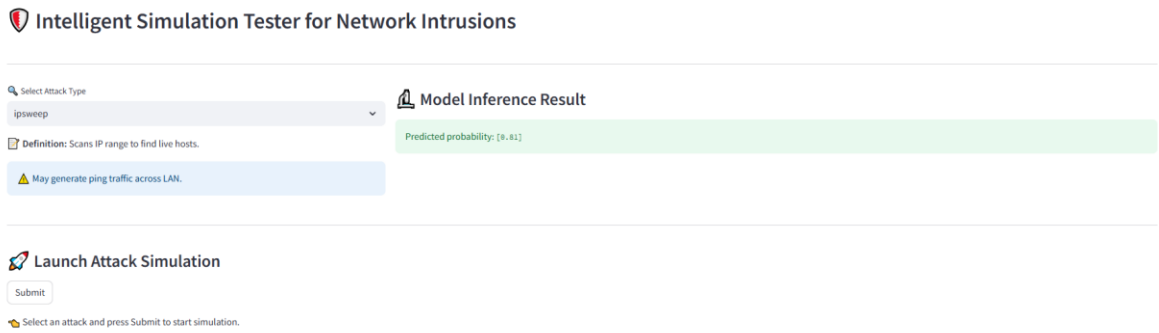


Figure 20: OS and Server-Agnostic Performance

The same test was executed on Ubuntu 20.04 and Ubuntu 16.04; the entire result along with the logs can be replicated. We even tested in different environments like Kaggle environment with and without GPU along with the colab environment and the entire platform along the results was replicated. This says that this python coding is agnostic of any kind of system settings

5.6 Real Time Scenario 3: In case the Network is Normal, any intrusion detected?

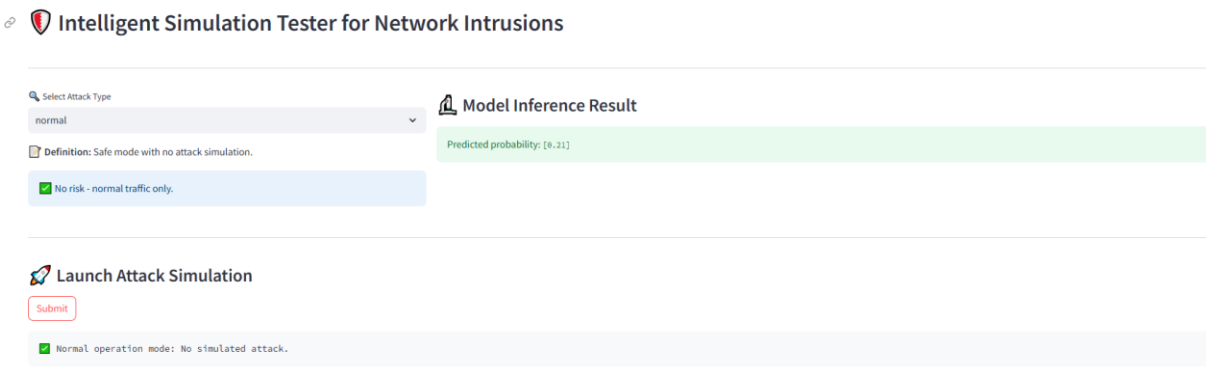


Figure 21: Model Response to Normal Traffic

The ML model being trained and deployed using the best detected algorithm is not detecting any kinds of suspicious activity and hence the system is normal as expected.

5.7 Real Time Scenario 4: In case of a vulnerable network, normal activity detected?

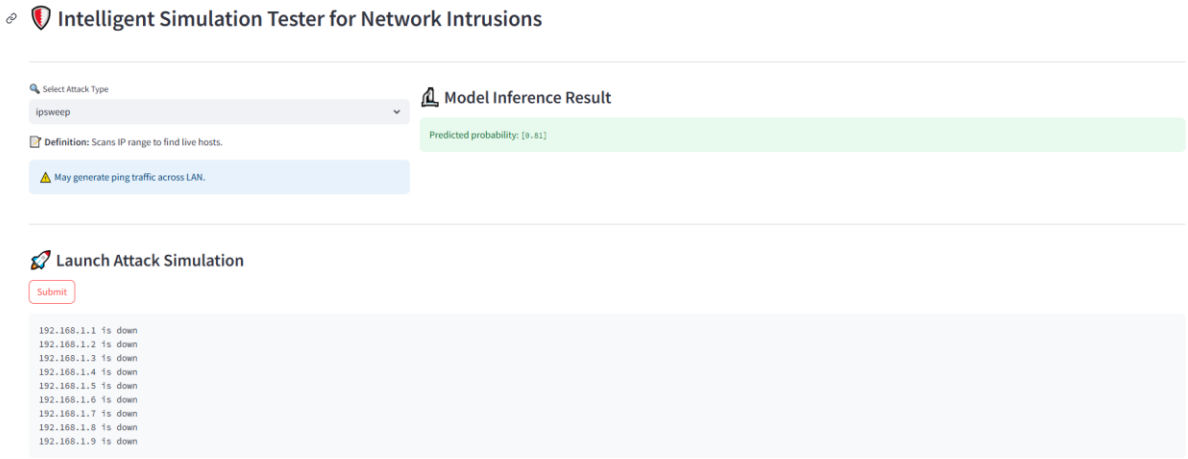


Figure 21: Response to an Abnormal behaviour

The model detects abnormal behavior and hence the same kind of mitigation to the attack is deployed to enhance the system's performance

Chapter 6: Conclusion and Future Work

By applying machine learning, this project proved that it is feasible to create a real time simulation system, which can be utilized in revealing abnormality as well as minimize vulnerability within network systems. The integration of the NSL-KDD dataset and binary classification models succeeded in taking the system to a terrific level of accuracy that is higher in identifying benign and malicious network activities. The use of static code analysis made it possible to identify software vulnerabilities effectively and as a result, the whole problem of security was seriously secured. Its stability was proven by its success in using a high degree of adaptability and platform-independence that allowed the work in a wide set of scenarios, including serverless, Linux, and Windows deployments among others, all in a real-time environment.

Our future plans are to apply deep learning models e.g. LSTM or Transformer architecture in order to boost system detection abilities. Temporal sequences are a rather important improvement because they allow these models to consider data on the networks. Two additional methods through which the model could be further modified to be more relevant to the current threat landscapes involve inclusion of threat intelligence data sources and usage of more current traffic behavior into the dataset. The second promising avenue to pursue is the automation of patch management (or other defensive actions) thereupon depending on the recognized threats. Finally, this system may be deployed on cloud-native systems, e.g. Kubernetes clusters, to allow distributed intrusion detection to be effectively scalable as well as lightweight in performance terms. In doing so, the system would be more appropriate to be utilized in businesses that have a wide scope of operations.

References

Abreu, D., Moura, D., Esteve Rothenberg, C. and Abelém, A., 2025. Quantumnetsec: Quantum machine learning for network security. International Journal of Network Management, 35(4), p.e70018.

Abreu, D., Rothenberg, C.E. and Abelém, A., 2024, June. QML-IDS: Quantum Machine Learning Intrusion Detection System. In 2024 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-6). IEEE.

Al-Shehari, T., Kadrie, M., Al-Mhiqani, M.N., Alfakih, T., Alsalman, H., Uddin, M., Ullah, S.S. and Dandoush, A., 2024. Comparative evaluation of data imbalance addressing techniques for CNN-based insider threat detection. Scientific Reports, 14(1), p.24715.

Balla, A., Habaebi, M.H., Elsheikh, E.A., Islam, M.R. and Suliman, F.M., 2023. The effect of dataset imbalance on the performance of SCADA intrusion detection systems. Sensors, 23(2), p.758.

Cultice, T., Onim, M.S.H., Giani, A. and Thapliyal, H., 2024, July. Anomaly detection for real-world cyber-physical security using quantum hybrid support vector machines. In 2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (pp. 619-624). IEEE.

De Ville, B., 2013. Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics, 5(6), pp.448-455.

Dong, J., Luo, B., Zhang, J., Zhang, P., Feng, F., Zhu, Y., Liu, A., Chen, Z., Shi, Y., Jiao, H. and Lu, G., 2025, March. Enhancing Large-Scale AI Training Efficiency: The C4 Solution for Real-Time Anomaly Detection and Communication Optimization. In 2025 IEEE International Symposium on High Performance Computer Architecture (HPCA) (pp. 1246-1258). IEEE.

Hdaib, M., Rajasegarar, S. and Pan, L., 2024. Quantum deep learning-based anomaly detection for enhanced network security. Quantum Machine Intelligence, 6(1), p.26.

Heese, R., Gerlach, T., Mücke, S., Müller, S., Jakobs, M. and Piatkowski, N., 2025. Explaining quantum circuits with shapley values: Towards explainable quantum machine learning. Quantum Machine Intelligence, 7(1), pp.1-33.

Inayat, U., Ayesha, R. and Mahmood, S., 2024, October. Improving Intrusion Detection System using Feature Extraction. In 2024 Horizons of Information Technology and Engineering (HITE) (pp. 1-6). IEEE.

Karim, H., Doshi, K. and Yilmaz, Y., 2024. Real-time weakly supervised video anomaly detection. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 6848-6856).

Kumar, R., Goswami, B., Mhatre, S.M. and Agrawal, S., 2024. Naive bayes in focus: a thorough examination of its algorithmic foundations and use cases. Int. J. Innov. Sci. Res. Technol, 9(5), pp.2078-2081.

Kunang, Y.N., Nurmaini, S., Stiawan, D. and Suprapto, B.Y., 2024. An end-to-end intrusion detection system with IoT dataset using deep learning with unsupervised feature extraction. International Journal of Information Security, 23(3), pp.1619-1648.

Kuznetsova, N., Westenberg, M., Buchin, K., Dinkla, K. and Van Den Elzen, S.J., 2014. Random forest visualization. Eindhoven University of Technology.

Lee, K.S., Kim, S.B. and Kim, H.W., 2023. Enhanced anomaly detection in manufacturing processes through hybrid deep learning techniques. IEEE Access, 11, pp.93368-93380.

Lopez-Ledezma, M. and Velarde, G., 2024, October. Cyber Security Data Science: Machine Learning Methods and Their Performance on Imbalanced Datasets. In International Scientific-Practical Conference (pp. 569-578). Cham: Springer Nature Switzerland.

Mahboubi, A., Luong, K., Aboutorab, H., Bui, H.T., Jarrad, G., Bahutair, M., Camtepe, S., Pogrebna, G., Ahmed, E., Barry, B. and Gately, H., 2024. Evolving techniques in cyber threat hunting: A systematic review. Journal of Network and Computer Applications, p.104004.

Mondragon, J.C., Branco, P., Jourdan, G.V., Gutierrez-Rodriguez, A.E. and Biswal, R.R., 2025. Advanced IDS: a comparative study of datasets and machine learning algorithms for network flow-based intrusion detection systems. Applied Intelligence, 55(7), p.608.

Nazat, S., Li, L. and Abdallah, M., 2024. XAI-ADS: An explainable artificial intelligence framework for enhancing anomaly detection in autonomous driving systems. Ieee Access.

Nemati, Z., Mohammadi, A., Bayat, A. and Mirzaei, A., 2024. Metaheuristic and data mining algorithms-based feature selection approach for anomaly detection. IETE journal of research, 70(7), pp.6040-6054.

Nguyen, M.T.A., Tong, V., Souihi, S.B. and Souihi, S., 2025. Zero Trust: Deep Learning and NLP for HTTP Anomaly Detection in IDS. IEEE Journal on Selected Areas in Communications.

Nick, T.G. and Campbell, K.M., 2007. Logistic regression. Topics in biostatistics, pp.273-301.

Peterson, L.E., 2009. K-nearest neighbor. Scholarpedia, 4(2), p.1883.

Rezaee, K., Rezakhani, S.M., Khosravi, M.R. and Moghimi, M.K., 2024. A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance. Personal and Ubiquitous Computing, 28(1), pp.135-151.

Rigatti, S.J., 2017. Random forest. Journal of insurance medicine, 47(1), pp.31-39.

Rohit, K., Kumari, P., Singh, N. and Alofaysan, H., 2025. Smart banking chatbots and consumer engagement: the role of trust and privacy in AI-driven banking. Journal of Strategic Marketing, pp.1-18.

Santa Barletta, V., Caivano, D., De Vincentiis, M., Pal, A. and Scalera, M., 2024. Hybrid quantum architecture for smart city security. Journal of Systems and Software, 217, p.112161.POLLARD, J., HALES, J., SHEN, Z. and DIXIT, V., A Novel Hybrid Quantum-Classical Framework for an In-Vehicle Controller Area Network Intrusion Detection.

Vishwanathan, S.V.M. and Murty, M.N., 2002, May. SSVM: a simple SVM algorithm. In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290) (Vol. 3, pp. 2393-2398). IEEE.

Yu, Y., Si, X., Hu, C. and Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), pp.1235-1270.