Anomaly Detection System based on Computing Intelligence

**1. Abstract:**

Our research relies entirely on artificial intelligence algorithms and machine learning models, namely stacking models. Stacking is an algorithm that takes the outputs of sub-models as input and tries to figure out how to combine the input predictions in the best way possible to get a superior output prediction.

Detecting and diagnosing the root cause of network traffic log problems is a time-consuming and labor-intensive process, especially for previously unknown failure modes. However, in the context of troubleshooting anomalies in network traffic logs, our project is based on a stacking method to identify malicious logs from the Advanced Security Network Metrics (ASNM) datasets.

According to the input training data, there have been roughly three orthogonal techniques to creating intrusion detectors: (1) Knowledge-based detection, which models and matches the characteristics of harmful intrusions. (2) Detection based on anomalies, which models typical behaviour and finds departures from it. (3) classification-based detection, which simultaneously models harmful and legitimate behaviour.

The drawbacks of these techniques include a high false negative rate in the case of evasions by unknown or zero-day attacks, a lengthy training and profiling process, and vulnerability.

To overcome all of the flaws, our project is entirely built on a stacking model, in which we took four machine learning algorithms and employed one of them at a time at level 1 and the other at level 0 for a higher testing accuracy rate. K-Nearest Neighbor, Nave Bayes, Support Vector Machine, and Decision Tree are the four methods employed.

The performance of these methods is quite comparable, with nave bayes being the most effective when retained at level 1, followed by support vector machine, Decision Tree, and K-Nearest Neighbor when kept at level 0.

**1.1 Introduction:**

To fight and manage cyberattacks and network security risks, developed and larger enterprises are strengthening their network security architecture these days. Because the risk of being hacked has grown considerably, even new technologies like cloud computing, fog, edge computing, and the Internet of Things (IoT) are vulnerable to such assaults.

Once these attackers breach network-related situations and get access to a company's or organization's network, the devastation they might inflict is unimaginable. They have the ability to do economic harm, steal sensitive information, and remain undetected in the network, causing more harm. Network anomaly detection systems (NADSs) are critical components of any network defence system since they analyse network packets for possible threats and aberrant user activity. Anomaly detection is the process of using algorithms to uncover abnormal or unexpected patterns in a network. Anomaly detection benefits an organisation in a number of ways, including informing them of their vulnerability to attacks and giving them time to change.

Any alteration in a network's unique established standard communication is referred to as an anomaly. Malware and cyberattacks, as well as inaccurate data packets and communication changes caused by network difficulties, capacity bottlenecks, or equipment failures, are examples of anomalies. Anomalies are identified in various companies using a variety of approaches, including: regular network intrusion checks, i.e., NIDS, abnormal finance activity detection, advanced penetration detection, and protecting web-based businesses.

**1.2 Real Life Implications:**

Artificial Intelligence is increasingly widely used in many aspects of life, resulting in more sophisticated life stories and better outcomes. AI structures assist in detecting and combating cyberattacks and cyber threats by utilizing a nonstop center of data, finding patterns, and retracing the assaults. AI is now being applied in almost every technology, including self-driving automobiles.

Machine Learning allows us to improve business choices, increase productivity, detect sickness, predict the weather, and much more. Essentially, a device learns from its inputs on a regular basis. Some of the cool gadget learning examples are traffic alerts, picture recognition, and so on.

**1.3 Related Works:**

**Paper 1: Dynamic Network Anomaly Detection System by Using Deep Learning Techniques.**

The author employed the LSTM (Long Short-Term Memory) strategy in the above-mentioned article. About LSTM = LSTM is a particular recurrent neural network structure suggested to handle the problem of long-term reliance. The forget gate instructs the neural network to forget the worthless information, the input gate instructs the neural network to add new content, and the output gate determines the current node's ultimate output.

In addition, he has made use of the Attention Mechanism. The Attention System (AM deep learning) is a simulation of the human brain's attention mechanism. When we read a piece of text, we usually focus on a few keywords so that we can quickly summarise the main content of the text; similarly, AM's role in anomaly detection is to calculate the impacts of each network traffic on the previous network traffic. The above-mentioned algorithm has a 93 percent accuracy. The author has employed a loss function to fine tune the algorithm's efficiency, which is one of the benefits of utilising the aforesaid technique (in the given function we calculate the loss).

**Paper 2: Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms**

Insider threats are security issues that emerge from people who have access to a company's network, systems, and data, such as workers and trusted partners. Insider threats are less often than external incursions, but the scale of harm is larger. The correctness of the suggested paper is 53.77 percent.

**Assumptions & data used:**

Individual user activity records that have been logged in the corporate system are gathered. Then, by describing individual actions, potential traits are extracted. If the system logs collect details on when a user connects his or her personal USB drive to the system, for example, the overall number of USB connections per day can be retrieved as an applicant variable, and subscriber contents, such as the body of an e-mail, can also be used to create candidate features.

**-Proposed method:**

After considering all of the aforementioned information, the author devised an algorithm for detecting insider risks in a company.

**Paper 3: A novel anomaly detection method based on adaptive MahalaNobis-squared distance and one-class KNN rule for structural health monitoring under environmental effects**

Unsupervised anomaly detection using Mahalanobis-squared distance (MSD) is a well-known method. Despite the MSD-based totally anomaly detection technique's widespread acceptance and wide applicability, a few key challenges and obstacles, such as climatic variables, commitment of a beyond the point threshold limit, prediction of an incorrect covariance matrix, and non-Gaussian of schooling numbers, can result in false alarms and inaccurate harm detection impacts. The primary purpose of this newsletter is to promote AMSD-KNN, an anomaly detection approach based entirely on adaptive Mahalanobis-squared distance and one-elegance KNN rule for SHM under diverse environmental conditions. The key idea behind the suggested method is to find adequate local friends of education and inspecting datasets in a -degree method for avoiding environmental variability situations and estimating nearby covariance matrices. To find enough nearest mates that ensure the estimate of well-conditioned neighbouring covariance matrices, a strong approach based entirely on a multivariate normality assumption check is provided. The suggested AMSD-KNN strategy is unique in that it uses a new multivariate distance degree and one-elegance KNN rule to build a unique unsupervised learning method for SHM. The block maxima (BM) approach is used to determine a reasonable threshold limit using specialized excessive cost distribution modelling. Due to the importance of picking excellent blocks inside the BM approach, a goodness-of-in-shape grade is calculated using the Kolmogorov-Smirnov hypothesis test to determine an optimal block number.

# The suggested methodologies' overall performance and efficacy are determined using well-known benchmark setups. A number of comparison studies are also carried out to demonstrate the superiority of the suggested procedures over a number of ultra-modern techniques.

# The suggested AMSD-KNN and BM approaches excel in detecting harm in low-variability environments, according to the results.

# **PAPER 4: Log-Based Anomaly Detection with the Improved K-Nearest Neighbor**

Log-based Anomaly Detection Using K-Nearest Neighbor Logs play a critical role in the safety of large-scale systems in this research article. The large range of logs that imply daily (everyday logs) differs greatly from the wide range of logs that suggest anomalies (bizarre logs), and the two types of logs have distinct characteristics. Detecting anomalies from logs manually using the K-Nearest Neighbor (KNN) set of rules, an outlier detection technique with high accuracy, is an effective way to find abnormalities. However, logs have the characteristics of a large scale and extremely choppy samples, which can influence the results of the KNN set of rules on log-based completely anomaly detection. As a result, we suggest a more advanced KNN set of rules-primarily based strategy that takes use of the current mean-shift clustering set of rules to successfully choose the educational set from large logs. Then, for samples with unique distances, we assign unique weights, which lessens the disastrous impact of unbalanced log statistical distribution at the efficiency of the KNN set of regulations. The results of experiments on log units from five supercomputers show that the approach we proposed can be efficiently applied to log-primarily based completely anomaly detection, and that the exactness, account fee, and F degree with our methodology are superior to those of the conventional key-word seek approach.

# **Paper 5: Deep Learning for Anomaly Detection**

# The goal of this research study's author is to give a full grasp of entirely anomaly detection techniques based on deep learning in a variety of software applications. To begin, it outlines the paradox detection problem, the tactics employed prior to the production of deep versions, and the difficult cases encountered. The book then examines today's deep mastery trends in depth, as well as the strategies used to overcome the restrictions imposed by traditional algorithms. It comes in second to last, looking into deep version anomaly detection techniques in real-world samples from LinkedIn production systems. The research concludes with a consideration of future trends. Then we realise how important it is to add modern deep anomaly detection algorithms. We discuss the following tasks in deep version anomaly detection strategies:

1) Using RNN, LSTM, Auto-Encoder, and other approaches, learn regular patterns from complicated data.

2) Detecting anomalies, in which we look at how to correctly detect anomalous behaviour using just reconstruction errors, reconstruction probabilities, and the use of a single class, KNN.

**Paper 6: Study and Analysis of Decision Tree Based Classification Algorithms**

Machine learning entails researching devices based on a variety of educational backgrounds, experimenting with statistics, and deciding the outcomes in each circumstance without the use of pre-programmed algorithms. Decision Tree is one of the device research methodologies. Decision Tree algorithms were employed in a variety of industries and for a variety of applications. These algorithms can be used to find statistics in other statistical procedures, to extract text, in scientific licenced sectors, and in search engines, among other things. Different decision tree algorithms have been developed based on their correctness and performance rate. It may be quite important for us to know how to apply a large range of rules in any case when we must make a decision. ID3, C4.five, and CART are three different Decision Tree methods presented in this study.

The dataset was subjected to the Decision Tree algorithms ID3 C4.five and CART. In terms of effectiveness, time, and accuracy, the decision tree exceeds the competition. It's built on a set of guidelines for locating interesting resources. Finally, the study of choice tree algorithms is concluded, and this research says that CART is a set of rules for this dataset that may be extremely particular and correct in comparison to others.

**Paper 7: An Improved KNN-Based Efficient Log Anomaly Detection Method with Automatically Labelled Samples**

The author of this study work concludes that outliers are logs that contain uncommon log states (anomaly logs), and that the k-Nearest Neighbor (KNN) set of rules has exceptionally high accuracy in outlier identification approaches. As a result, we employ the KNN set of rules to find abnormalities within the log data. However, there are a few issues with using the KNN set of rules to find anomalies, three of which are: excessive vector measurement results in inefficient KNN set of rules, unlabelled log information is useless to the KNN set of rules, and the imbalance of the range of log information distorts the type selection of the KNN set of rules. We offer a green log anomaly detection solution based entirely on a stepped forward KNN set of rules with a mechanically classified pattern set to solve those three issues. This method presents a log parsing method based only on N-grams and a common sample mining (FPM) method that decreases the measurement of the log vector modified with Frequency distribution. Inverse Document Frequency (TF-IDF) is a technique that uses inverse document frequency. Then, using clustering and self-schooling, we automatically extract classified log information patterns from old logs.

For odd logs with tiny amounts and long distances from conventional logs, we employ a clustering and self-training technique to provide classified log statistics pattern set on a regular basis. Finally, we apply common weighting distance to improve accuracy of the KNN algorithm, reducing the detrimental effects of log pattern imbalance. The results show that our approach can improve the effectiveness of log-based totally aberration detection with the KNN algorithm while ensuring accuracy at the same time, based on experiments on log units generated through six datasets of various types and comparisons with three different log-based totally anomaly detection methods.

**Paper 8: A Lightweight Anomaly Detection Model using SVM for WSNs in IoT, through a Hybrid Feature Selection Algorithm based on GA and GWO**

Anomaly or intrusion detection system (IDS) is an effective defence mechanism for stressed networks. However, modern technology with high computational complexity is mistaken for aid-restricted WSNs in IoT, and they also fail to identify new WSN assaults. Managing the large number of incursions wi-fi site visitors obtained by sensors, imposing a gradual detecting procedure, improved assistance use, and inaccurate detection As a result, considering WSN hurdles for developing an IDS in the IoT is a significant challenge for security researchers. This study offers a novel version of the GAB GWO to improve a support vector machine (SVM)-based completely light-weight IDS (LIDS) by combining concepts from genetic algorithms (GA) with mathematical equations from the grey wolf optimizer (GWO). The GABGWO uses novel crossover and mutation operators to try to find the most useful site visitor functions and eliminate the useless ones, in order to improve the LIDS' overall performance. The overall performance of LIDS is assessed using the AWID real-world wi-fi dataset in scenarios with and without the use of GAB GWO. The results supported the suggested GAB GWO set of rules' promising performance in choosing on the most desirable traffics, lowering computing expenses, and delivering excessive accuracies for LIDS. The hybrid set of rules has been compared to natural GA and GWO, as well as other recent approaches, and it has been established that its overall performance is superior.

**Paper 9: Study and Analysis of Decision Tree Based Classification Algorithms**

Deep learning is the technique of analysing devices based on a variety of academic and testing information and deciding the outcomes in different situations without being explicitly programmed. One of the device learning approaches is the Decision Tree. Decision Tree algorithms have been used in a diverse range of companies and applications. These methods may be used to extract text, clinically licenced fields, and search engines, as well as identify statistics in a number of statistical activities. On the basis of their accuracy and efficiency rate, many decision tree algorithms were constructed. It may be critical for us to understand how to employ a good set of guidelines in any decision-making situation. In terms of accuracy, time, and precision, the decision tree exceeds the competition. It is based on a set of rules that are used to provide recommendations for finding interesting sites. Finally, a thorough examination of choice tree algorithms is completed, and this study finds that CART is a set of rules for this dataset that is more specific and accurate than many others.

# **Paper 10: A Novel Anomaly Detection Algorithm Using DBSCAN and SVM in Wireless Sensor Networks**

The author discusses the fact that such networks cannot be supervised, and hence this research addresses the problem of anomaly detection. First, the community traffic is used to extract the three functions of temperature, humidity, and voltage. The density-primarily based fully spatial clustering of packages with noise (DBSCAN) collection of guidelines is then used to cluster community information. It also uses density-based totally detection techniques to assess the correctness of the DBSCAN set of criteria for entering information. This collection of criteria identifies variables in low-density areas as anomalous. It trains to assist vector machines by using daily information. Finally, it eliminates outliers from community data. The suggested set of rules is examined using Intel Berkeley Research lab's normal and standard facts set (IRLB). Using coefficient correlation, we should be able to solve DBSCAN's problem of deciding on entry parameters in this study. The suggested set of rules has an advantage over previous ones in that it uses gentle computing methods, is simple to implement, and improves detection accuracy by evaluating these three functions simultaneously.

**Paper 11: On-Line Anomaly Detection with High Accuracy**

The author of this paper highlights how traffic anomaly detection is important for better Internet administration. Traditional detection algorithms frequently transform excessively dimensional recordings to a prolonged vector, which reduces detection accuracy due to a lack of spatial analysis in the records. Furthermore, they're frequently created based wholly on the segregation of regular and abnormal records in a time period, which not only adds extra trash and computing price, but also inhibits timely identification of anomalies. It is critical to discover online and fix website navigation anomalies, but it is difficult to do so. To deal with the problem, this research creates a 2-D matrix from the monitoring records in each time slot and uses bilateral big thing analysis to discover anomalies inside the new time frame (B-PCA). We recommend a number of novel strategies in Online B-PCA to aid quick and accurate anomaly detection in real time, including a unique B-PCA-based totally anomaly detection precept that considers the variant of each row and column major instructions for more accurate anomaly detection, an approximate set of rules to avoid using the generation process to calculate the major instructions in a close-form, and a sequential anomaly set of rules. That is, to the best of our knowledge, the first artwork to use 2-D PCA for anomaly detection. We ran massive simulations to test our Online B-PCA using state-of-the-art anomaly detection techniques and real-world site visitor strains Abilene and GANT. Our simulation results demonstrate that, when compared to other algorithms, our Online B-PCA can achieve much better overall detection performance with a low fake effective rate, a high legitimate effective rate, and an excessive caffeine calculation value.

**1.3 Proposed Solution:**

In this paper, we offer a new method for determining if a network request has harmful intent. It contains a variety of machine learning modules and approaches such as stacking. In this model, we integrate several algorithms such as KNN, SVM, Naive Bayes, and decision trees to fit them into different tiers of the stack, such as tiers 0, 1, and 2. After that, we execute the code on the data sets to ensure that it is as accurate as possible in all three datasets.

**1.4 Research Contribution**

The strategy we utilized, namely the stacking model, is effective and has a low false alarm rate. The contribution of our work in the field of anomaly detection is that our algorithm can help organizations understand anomalous behavior in their networks. One of the benefits of knowing the anomaly in the network is that the organization can work on it and become immune to attacks or breaches via that particular anomaly. Also, we have decided to focus more on the algorithm's false alarm rate, as a massive proportion of these can cause problems.

**2. Materials and Method**

**2.1 Experimental Methods:**

The following are the prerequisites for the machine that will be used to carry out the research:

**Table 1.** Tested Environment

|  |  |
| --- | --- |
| OS | Windows 10 |
| RAM | 8 GB |
| GPU | 4 GB |
| IDE | Visual Studio Code (Python) |

The aforementioned setup is used to implement the complete algorithm. The physical requirements are minimal, as shown in the table above, and the suggested model may be used with practically any physical machine.

**2.2 Data Input:**

We'll start by importing all of the necessary libraries, which are as follows:

**1. Pandas:** Pandas is an open-source Python package that is extensively used for data analysis and machine learning activities; it is utilized for data cleaning and analysis in this case.

**2. NumPy:** A Python library that adds support for massive, multi-dimensional arrays and matrices, as well as a vast set of high-level mathematical functions to work on them.

**3. Sklearn:** is a popular tool for statistical modelling, such as classification, regression, clustering, and dimensionality reduction.

**4. Matplotlib:** Matplotlib is a numerical extension of the NumPy library for cross-platform data visualization and graphical charting in Python.

**2.3 Pre-Processing:**

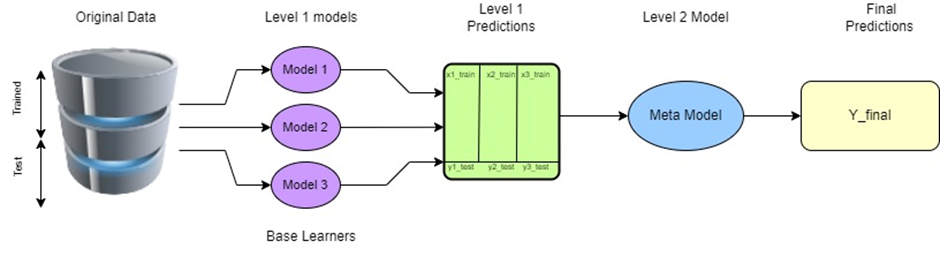
**2.3.1 Importing the Dataset:** The ASNM dataset has three sub-datasets: ASNM-CDX-2009, ASNM-NBPOv2, and ASNM-TUN. Following that, we'll import them one by one for each of the 12 programs.

**2.3.2 Segmentation and Masking:** Because our dataset is unorganized, we need to clean it up by separating and labelling features and storing them in two distinct arrays.

**2.3.3 Resizing:** All network traffic data in the form of IP addresses, MAC addresses, port numbers, and VLAN IDs is converted to binary form for more accurate computation and display.

**2.3.4 Standardization:** The CSV dataset's entries were all subjected to standardization, resulting in an average of 0 and a unit standard deviation can be seen.

**2.4 Proposed Model Architecture:**

** Figure. 1. The Stacking Method Architecture Diagram**

**2.5 Data Source**

The publicly available data source comes from the IT4Innovations Center of Excellence, Faculty of Information Technology, Brno University of Technology, 612 00 Brno, Czech Republic, and consists of three datasets built from network traffic traces using ASNM (Advanced Security Network Metrics) features. There are 5000-6000 rows and 50-60 columns in each of these datasets.

There are three sub-datasets in our dataset:

***(i.) ASNM-CDX 2009:*** This dataset consists of ASNM features extracted from tcp-dump capture of malicious and legitimate TCP communications on network services which are vulnerable to buffer overflow attacks and are included in CDX-2009 dataset of network traffic dumps.

The final composition of the dataset is depicted in table ASNM-CDX-2009 dataset contains two types of labels that are enumerated by increasing order of their granularity in the following:

. **Label\_2:** Is a two-class label, which indicates whether an actual sample represents a network buffer overflow attack or legitimate traffic.

**. Lebel\_poly:** is composed of two parts that are eliminated be a separator:

**(a)** A two-class label where legitimate and malicious communications are represented by symbols 0 and 1 respectively.

**(b)** An acronym of network service. This label represents the type of communication on a particular network service.

***(ii.) ASNM-NPBO v2:*** This dataset contains non-payload-based obfuscation techniques applied onto malicious traffic and onto several samples of legitimate TCP communications on selected vulnerable network services. The selection of vulnerable services was aimed on high severity of their successful exploitation leading to remote shell code execution through established backdoor communication.

legitimate representatives of the dataset were collected from two sources:

a)  Legitimate traffic simulation in our virtual network architecture and also employed non-payload-based obfuscations for the purpose of real network simulation.

b)  Common usage of all selected services was captured in the campus network, and all traffic was anonymized and further filtered on high severity alerts by signature-based NIDS Suricata and Snort through virus total API.

***(iii.) ASNM-TUN:*** It consists of ASNM features extracted from tcp-dump capture tunnelling obfuscation techniques applied onto malicious traffic created with the intention to evade and improve machine learning classifiers and besides legitimate network traffic samples.

ASNM-TUN dataset contains four types of labels that are listed by increasing order of their level in the following:

· **Label\_2:** It is a two-class label, which indicates whether an actual sample represents a network buffer overflow attack or a legitimate communication.

· **Lable\_3:** It is a three-class label, which distinguishes among legitimate traffic, direct attacks and obfuscated network attacks.

· **Label\_poly:** It is a label that is composed of two parts:

**(a)** A three-class label

**(b)** An acronym of a network service

· **Label\_poly\_s:** It is composed of three parts:

**(a)** A three-class label

**(b)** An acronym of network service

**(c)** A network modification technique involved.

**2.6 Parameters and customization for computer visions Models**

Table 1 shows the various parameters setting for Computer Vision Models.

***2.6.1 Stacking Model:***

Stacking is an ensemble learning strategy that generates predictions via meta-learning. The original data is split into n-folds (train data and test data), and these data are then integrated into various models (we utilized KNN, SVM, Nave bayes, and Decision tree models in this study) to provide predictions, which could then be added back to the level 2 to produce a final conclusion.

***2.6.2: K Nearest Neighbor (KNN):***

The K Nearest Neighbor method is a classification algorithm that falls under the supervised learning category. It uses K Nearest Neighbors (Data Points) to estimate the new datapoint's class or continuous value.

***2.6.3: Support Vector Machine (SVM):***

Support Vector Machine is a supervised machine learning technique that may be used to solve problems in classification and regression. We depict each data item as a point in N-Dimensional space (where N is the number of features you have) with the value of each feature being the value of a certain coordinate in this algorithm. Then we classify the data by locating the hyper-plane that separates the two groups.

***2.6.4: Naïve Bayes:***

The existence of a given characteristic in a class is independent to the presence of any other feature, according to the Nave Bayes classification approach, which is simple to create and particularly helpful for huge data sets.

***2.6.5 Decision Tree:***

For classification and regression issues, decision trees can be employed. It employs a flowchart that looks like a tree structure to depict the predictions that arise from a sequence of feature-based splits, as the name implies. It begins with a root node and finishes with a leaf decision.

**3. Results and Discussions:**

The train-test split method was used to divide the dataset into two separate subarrays. The training component accounted for 75% of the original data, whereas the testing component made up 25% of the original dataset. Following the processing of all of the models' details, a performance-based inquiry was undertaken to choose the best model out of four. The parameter metrics used to identify models in this comparative analysis were accuracy, precision, F1 score, and recall, which may be computed from the confusion matrix. The likelihood of belonging to a certain class is used to represent all of the models' output. Which can take the shape of a fraction ranging from 0 to 1.

There are three datasets in this example:

· ASNM-CDX 2009 dataset

· ASNM-NBPOv2 dataset

· ASNM-TUN dataset

And four algorithms are being used and they are as follows:

· K Nearest Neighbor (KNN)

· Decision Tree (DT)

· Support Vector Machine (SVM)

· Naïve Bayes (NB)

We employed a stacking strategy to identify the best algorithm for finding improved accuracies in our dataset, therefore we have four programs for each dataset, and three datasets in total. For each dataset, the four programs are divided as follows:

**1. Level 0:** DT, SVM, NB **Level 1:** KNN

**2. Level 0:** KNN, SVM, NB **Level 1:** DT

**1. Level 0:** KNN, DT, NB **Level 1:** SVM

**1. Level 0:** KNN, DT, SVM **Level 1:** NB

**Figure. 2.** Comparison of metrics – (Precision, Recall, F1-score) for ASNM-CDX 2009 Dataset

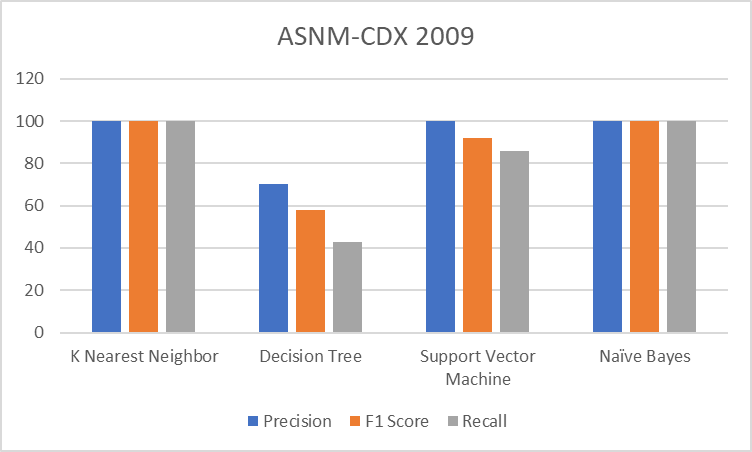
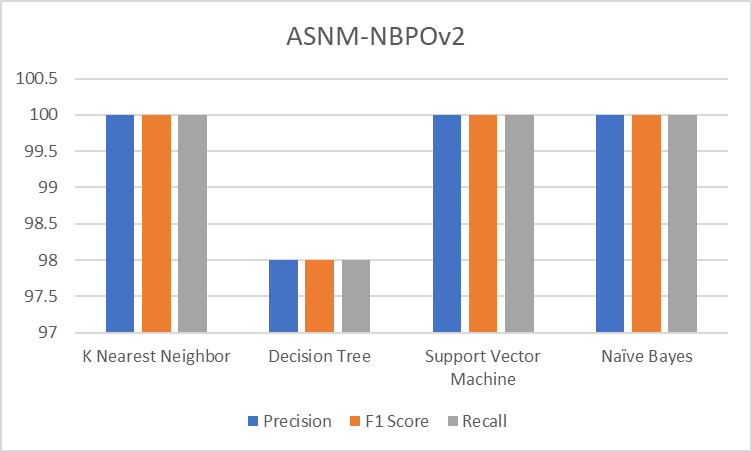


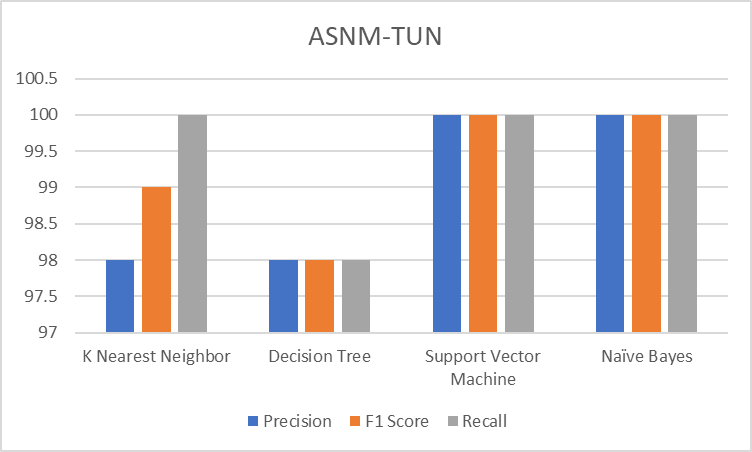
Figure 2. depicts the performance shown by four algorithms on ASNM-CDX 2009 dataset. K Nearest Neighbor algorithm and Support Naïve Bayes algorithm have a perfect score of metrics. The Support Vector Machine model has a high precision score only while least metrics are shown by the decision tree model.

**Figure. 3** Comparison of metrics – (Precision, Recall, F1-Score) for ASNM-NBPOv2 Dataset.



The graph of figure 3 shows the performance of all AI models on ASNM-NBPOv2 dataset. Support Vector Machine, Naïve Bayes and K Nearest Neighbor show high values in all the three metrics while Decision Tree accounted for 98%.

**Figure. 4** Comparison of Metrics – (Precision, Recall, F1-Score) for ASNM-Tun dataset



The bar graph of figure 4 shows the performance of the 3 Artificial Intelligence models on ASNM-TUN dataset. Support Vector Machine and Naïve Bayes display all the metrics with utmost percentage. The Precision of K Nearest Neighbor is 100% while the decision tree has equal AMOUNT OF proportion in all the three domains.

**Table. 2.** Confusion Matrix of the **K-Nearest Neighbor Algorithm** with all three datasets of ASNM dataset at level 1.

**1.** **For ASNM-CDX 2009 Dataset:**

|  |  |
| --- | --- |
| 1429 | 0 |
| 0 | 14 |

**2.** **For ASNM-NBPOv2 Dataset:**

|  |  |
| --- | --- |
| 2691 | 2 |
| 2 | 167 |

**3.** **For ASNM-TUN dataset:**

|  |  |
| --- | --- |
| 43 | 0 |
| 0 | 55 |

**Table. 3.** Confusion Matrix of **Decision Tree Algorithm** with all three datasets of ASNM dataset at level 1.

**1.** **For ASNM-CDX 2009 Dataset:**

|  |  |
| --- | --- |
| 1425 | 4 |
| 8 | 6 |

**2.** **For ASNM-NBPOv2 Dataset:**

|  |  |
| --- | --- |
| 2689 | 4 |
| 3 | 166 |

**3.** **For ASNM-TUN dataset:**

|  |  |
| --- | --- |
| 42 | 1 |
| 2 | 53 |

**Table. 4.** Confusion Matrix of **Support Vector Machine Algorithm** with all three datasets of ASNM dataset at level 1.

**1.** **For ASNM-CDX 2009 Dataset:**

|  |  |
| --- | --- |
| 1429 | 0 |
| 0 | 14 |

**2.** **For ASNM-NBPOv2 Dataset:**

|  |  |
| --- | --- |
| 2693 | 0 |
| 0 | 169 |

**3.** **For ASNM-TUN dataset:**

|  |  |
| --- | --- |
| 43 | 0 |
| 0 | 55 |

**Table. 5.** Confusion Matrix of the **Naïve Bayes Algorithm** with all three datasets of ASNM dataset at level 1.

**1.** **For ASNM-CDX 2009 Dataset:**

|  |  |
| --- | --- |
| 1429 | 0 |
| 2 | 12 |

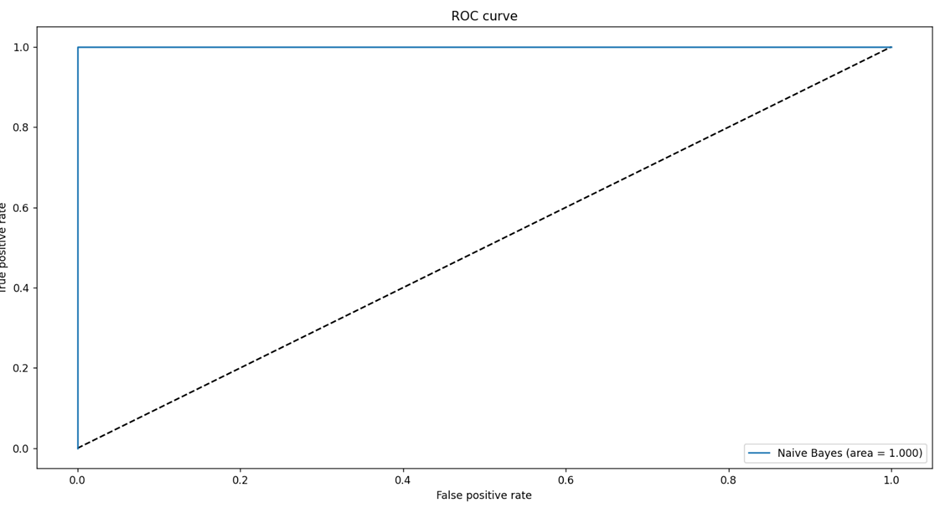
**2.** **For ASNM-NBPOv2 Dataset:**

|  |  |
| --- | --- |
| 2693 | 0 |
| 0 | 169 |

**3.** **For ASNM-TUN dataset:**

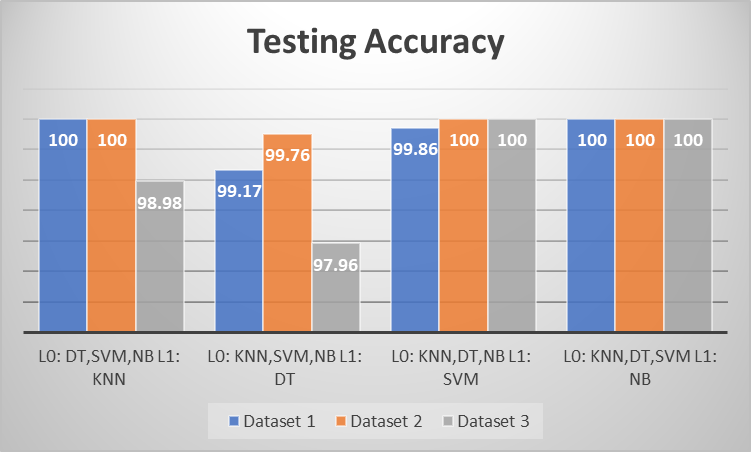
|  |  |
| --- | --- |
| 43 | 0 |
| 0 | 55 |

**Figure. 5. ROC curve for KNN, Decision tree, SVM at level 0 and naïve bayes at level 1 model.**

****

The above model of KNN, SVM and Decision Tree at level 0 and Naïve Bayes at level 1 model shows utmost true positive to false positive ratio with area under curve of 1.000.

After a through comparison and contrast, it is empirical that the proposed model of Naïve Bayes at level 1 and other three algorithms at level 0 has surpassed all the other techniques, with an accuracy of 100%, and the side metrics of Precision, F1 Score and Recall score were found to be 100% in all the sub dataset of ASNM dataset. **(Figure 6)**.



From the above graph it is clearly visible that the model in which Naïve Bayes is at level 1 has 100% accuracy in all the three datasets.

Anomalies being a disturbance to any result or work make the output deviate between observed data and the normal state but with the help of this project idea we will be able to identify the best algorithms for anomaly detection with highest accuracy by using Machine learning and deep learning models we can implement it and

**4. Conclusion:**

In the end, the outlier analysis revealed that the hybrid model of algorithms for log detection has a greater accuracy percentage than the traditional anomaly detection approach, and hence produces superior outcomes. After integrating multiple methods, we found that **L0: KNN, DT, SVM, and L1: NB** this hybrid branch model has the greatest accuracy and will thus be selected for resolving difficulties in datasets/logs for spotting these uncommon patterns, i.e., anomalies.

**4.1** **Future Work:**

We may use our study paper's approach in future projects to improve the accuracy and precision of anomaly detection in data sets. In comparison to solitary algorithms for anomaly detection, the hybrid model produces superior outcomes and accuracy.

By enhancing the accuracy of our algorithm, we may improve the accuracy of our findings, resulting in findings that are more accurate and exact. Projects will be more accurate and trustworthy as a result of this.

**5. References:**

1. <https://ieeexplore.ieee.org/document/9115004>
2. <http://www.fit.vutbr.cz/~ihomoliak/asnm/>
3. <https://www.edureka.co/blog/what-is-a-neural-network/>
4. <https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207#:~:text=Artificial%20Neural%20Network(ANN)%20uses,complex%20patterns%20and%20prediction%20problems>.
5. <https://www.ijitee.org/wp-content/uploads/papers/v8i9/I7914078919.pdf>
6. <https://www.mdpi.com/1099-4300/23/5/529>
7. <https://www.analyticssteps.com/blogs/8-applications-neural-networks>
8. <https://www.xenonstack.com/blog/artificial-neural-network-applications>
9. <https://towardsdatascience.com/building-our-first-neural-network-in-keras-bdc8abbc17f5>
10. <https://link.springer.com/article/10.1007/s11277-017-4961-1>
11. <https://www.worldscientific.com/doi/abs/10.1142/S0218194020500114>
12. <https://dl.acm.org/doi/abs/10.1145/3336191.3371876>
13. <https://dl.acm.org/doi/abs/10.1145/3441448>
14. <https://www.researchgate.net/profile/Purvi-Prajapati/publication/330138092_Study_and_Analysis_of_Decision_Tree_Based_Classification_Algorithms/links/5d2c4a91458515c11c3166b3/Study-and-Analysis-of-Decision-Tree-Based-Classification-Algorithms.pdf>
15. <https://jcomsec.ui.ac.ir/article_24558_4491.html>
16. [(PDF) Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms (researchgate.net)](https://www.researchgate.net/publication/336065245_Insider_Threat_Detection_Based_on_User_Behavior_Modeling_and_Anomaly_Detection_Algorithms)
17. <https://www.researchgate.net/publication/342118050_ASNM_Datasets_A_Collection_of_Network_Attacks_for_Testing_of_Adversarial_Classifiers_and_Intrusion_Detectors>
18. <https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2109>
19. <https://link.springer.com/chapter/10.1007/978-3-642-31537-4_46>
20. <https://www.researchgate.net/publication/347635021_Cyberattacks_Detection_in_IoT-Based_Smart_City_Applications_Using_Machine_Learning_Techniques>
21. <https://www.irjet.net/>
22. <https://www.researchgate.net/publication/336767849_ASNM_Datasets_A_Collection_of_Network_Traffic_Features_for_Testing_of_Adversarial_Classifiers_and_Network_Intrusion_Detectors>