Literature Review

Introduction:

In today's interconnected digital landscape, the prevalence of cyber threats has escalated, necessitating robust security measures to safeguard sensitive information and maintain system integrity. Automated Intrusion Detection Systems (IDS) leveraging machine learning offer a proactive solution to this challenge. These systems are designed to identify and mitigate unauthorized access attempts and other malicious activities by analyzing patterns and anomalies in network traffic and system behavior.

The necessity for such systems arises from the increasing sophistication of cyber attacks, which often elude traditional rule-based security measures. Machine learning algorithms enhance IDS capabilities by continuously learning from new data, adapting to evolving threats, and reducing false positives through advanced pattern recognition.

Paper Review

Paper 1
Title: Overview and Exploratory Analyses of CICIDS 2017 Intrusion Detection Dataset
Authors: Oyelakin A. M, Oyelakin A. M et.al

This study explores the CICIDS2017 dataset and it's effectiveness in designing an intrusion detection system. It explores the added attacks that can be detected using this dataset and some of the shortcomings associated with this dataset. It also explores the features present in the dataset and allows for better feature engineering. It also provides an insight to relabel the dataset to address the extreme class imbalance present in this dataset. The problems highlighted by this paper are as follows:

1. Missing values
2. Class Imbalance
3. Huge Volume of Data
4. Scattered presence of data

It also explored the features present in the dataset and the relevance of each feature. With proper visualization and exploration, this paper concludes that this dataset requires an innovative approach during the pre-processing phase so that we can build more effective intrusion detection models.

Paper 2:
Title: An intrusion detection system empowered by deep learning algorithms
Authors: Weiye Tang, Daren Li, Wuyue Fan, Tianhao Liu, Man Chen, Omar Dib

The Industrial Internet of Things (IIoT) requires robust cybersecurity measures to manage increasing network traffic and protect against cyber attacks. Traditional machine learning techniques struggle to capture complex patterns in the vast data from IoT devices. Deep learning (DL), with its multi-layer neural networks, improves accuracy and generalization by extracting hierarchical features from data. This paper examines recent DL and optimization techniques for IIoT intrusion detection, aiming to develop more accurate, robust, and efficient threat detection models. Experiments on the EdgeIIoT dataset show high accuracy and rapid responsiveness, validating the proposed IDS system's feasibility.

Datasets Used:

1. Edge IIoT: The Edge IIoT (Industrial Internet of Things) dataset is a comprehensive collection of data specifically designed to support research and development in the field of IIoT. It includes various types of data collected from industrial environments, which can be used for tasks such as anomaly detection, predictive maintenance, and optimization of industrial processes. The dataset typically contains:

    a. **Sensor Data**: Measurements from various industrial sensors, including temperature, pressure, humidity, and vibration.
    b. **Event Logs**: Records of specific events or anomalies detected in the system.
    c. **Machine Status**: Information on the operational status of machinery, such as active, idle, or under maintenance.
    d. **Maintenance Records**: Historical data on maintenance activities performed on industrial equipment.
    e. **Production Data**: Information related to the production output, quality control, and operational efficiency.

Models Used:
In this paper, they convert the tabular data into image representation so as to apply CNN. They use algorithms like AutoKeras, H2O, EfficientNetV2, Vision Transformer and InceptionResNetv2. A brief description of each of the algorithms are as follows:

a. AutoKeras: AutoKeras is an open-source software library designed to automate the process of building deep learning models. It provides an easy-to-use interface for developing neural networks, allowing users to automatically search for the best architecture and hyperparameters for their specific tasks, such as image classification, text classification, and regression. AutoKeras is built on top of Keras and TensorFlow, making it accessible for both beginners and experienced machine learning practitioners.
b. H2O: H2O is an open-source machine learning platform that supports a range of algorithms, including deep learning. In the context of deep learning, H2O provides a user-friendly interface for building and training neural networks, integrating with popular frameworks like TensorFlow and Keras. It offers scalable and efficient tools for model

development, hyperparameter tuning, and deployment, making it suitable for large-scale datasets and complex tasks.

c. EfficientNetV2 is a family of convolutional neural network architectures designed for efficient performance in image classification tasks. Developed by Google, it builds upon the original EfficientNet by introducing improvements in both model scaling and training efficiency. EfficientNetV2 achieves better accuracy and faster training times through techniques like compound scaling, which balances network depth, width, and resolution, and the use of optimized training procedures and data augmentation strategies.
d. Vision Transformer: The Vision Transformer (ViT) is a deep learning model for image classification that leverages transformer architecture, originally developed for natural language processing. Instead of using convolutional layers, ViT processes images by dividing them into fixed-size patches, linearly embedding these patches, and then applying self-attention mechanisms to capture global dependencies. This approach enables ViT to achieve state-of-the-art performance on image classification tasks by leveraging the transformer's ability to model long-range relationships within the data.
e. InceptionResNetv2: Inception-ResNetV2 is a deep convolutional neural network architecture that combines the strengths of the Inception module with residual connections. It improves upon the original Inception architecture by incorporating residual connections, which help in training deeper networks by alleviating the vanishing gradient problem. The network uses a series of Inception modules for feature extraction and residual blocks to facilitate gradient flow and enhance performance. Inception-ResNetV2 is known for its efficiency and effectiveness in image classification tasks.

Results obtained:
The results obtained are shown in the table below:

Method	Acc(%)	DR(%)	F1(%)
H2O	99.99%	99.99%	99.99%
Autokeras	99.82%	99.82%	99.82%
EfficientNetV2	99.85%	99.84%	99.84%
Vision Transformer	99.71%	99.69%	99.70%
InceptionResNetV2	99.58%	99.58%	99.58%
MLP [23]	93.64%	94.78%	93.77%
Centralized Blending [23]	90.04%	92.34%	90.54%

The models perform extremely well in most of the cases but these models need to be implemented in an IoT setting which may not be feasible. Proper consideration must be performed in this case.

Paper 3
Title:Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network

Author: EMRAH TUFAN, Cihangir Tezcan, Cengiz Acartürk

The study explores network intrusion attempts using anomaly-based machine learning models, aiming to enhance protection compared to traditional misuse-based models. Two models were developed and implemented: an ensemble learning model and a convolutional neural network (CNN) model. These models were tested on a dataset collected from a real-life, institutional production environment. To verify their reliability and validity, the models were also applied to the UNSW-NB15 benchmarking dataset. The focus was narrowed to probing attacks to maintain a manageable scope. The results showed high accuracy rates, with the CNN model being slightly more accurate.

Datasets Used

1. The Institutional Network datasets:

   The data for this study was collected from an organization's network using a tool called tcpdump. The data collection focused on sessions initiated from outside sources, captured from a specific network interface. Data was gathered during both working hours and off-hours, resulting in five different captures. After processing, the final dataset included 100,000 flows or sessions, randomly sampled from an initial 481,000 sessions. These sessions involved 38,804 unique hosts, but most of the traffic came from a few clients, with the top 20 clients accounting for 36% of all sessions. Similarly, the top 5 destination hosts received 57% of the sessions. The most common applications were HTTP, DNS, telnet, SMB, SQL, SMTP, and SSH, making up 55% of the sessions. On average, each session lasted 6.06 seconds, transferred 41,385 bytes, and involved 62 packets.

2. UNSW-NB15 dataset

   The UNSW-NB15 dataset was created by the cybersecurity research group at the Australian Center for Cyber Security (ACCS). It includes simulated attacks grouped into nine categories: fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms. These attacks were designed using information from a common vulnerabilities and exposures (CVE) website. Tcpdump was used to capture the traffic, resulting in a large dataset of 2,500,000 flows or sessions. For this study, 100,000 flows were randomly selected for analysis, focusing on probing attacks to keep the study manageable and avoid accessing sensitive information. Probing analysis was done

using metadata and features extracted from various network layers. The same approach was used for the UNSW-NB15 dataset, simplifying it to a binary format with probing and non-probing classes.

Pipeline

1. **Data collections and formatting**
   After capturing network traffic, the data was divided into segments of 2 million packets for easier processing. Network address translation (NAT) IP addresses were converted to public addresses using the tcprewrite tool, ensuring consistency for analysis tools like argus and tcptrace. The captured traffic was in Linux cooked capture format, which lacks layer 2 (data link layer) addresses. To properly analyze this data with tools like Wireshark and argus, pseudo-random MAC addresses were assigned using tcprewrite. This arbitrary assignment did not introduce bias since no features were extracted from layer 2 in the models.

2. **Feature Extraction**
   Building a flow-level intrusion detection model is more common and efficient than a packet-level model because it requires less computational power and time. For instance, 250 MB of packet data includes about 2 million packets but only around 15,000 flows. Additionally, some attacks are only detectable when evaluating packets within their sessions. Using tcptrace and argus, we extracted flow-level features from packet-level data. Custom features were created to detect anomalies based on source behavior over time, using tools like Pyshark, pandas, and numpy. We focused on specific TCP header flags indicating probing behavior, such as ICMP and syn-ack. These features were combined, using session start time as the index, but we had missing values for non-TCP flows like UDP and ICMP, which were addressed later in the missing-value handling phase.

3. **Models**
   - CNN
   - Ensemble learnings

   In this study, we developed two machine learning models for anomaly-based intrusion detection: an ensemble learning model and a convolutional neural network (CNN). The ensemble model used the bagging method with naive Bayes, KNN, logistic regression, and SVM as base classifiers. The CNN model converted feature vectors into 2D images for input. Both models were trained and tested on our institutional data set and the UNSW-NB15 data set. We used the hyperas library for hyperparameter optimization and applied techniques to prevent overfitting, such as regularization with dropout. The data sets were split into training, validation, and testing sets, and the models were evaluated for accuracy.

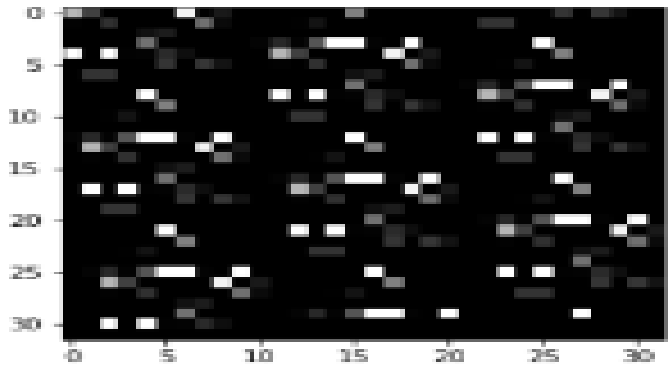


Fig: An example flow which was converted to 32 × 32 grayscale Image.

Image Size	F1 Score	ROC AUC Score	Train/Test Duration for 10000 Samples	Data Size
16 x16	0.9818	0.9878	154 seconds	20 MB
32 x 32	0.9911	0.9927	343 seconds	80 MB
64 x 64	0.9929	0.9944	25 minutes	330 MB
128 x 128	0.9943	0.9958	114 minutes	1.4 GB

Fig: Summary of results from various image sizes for 2D conversion.

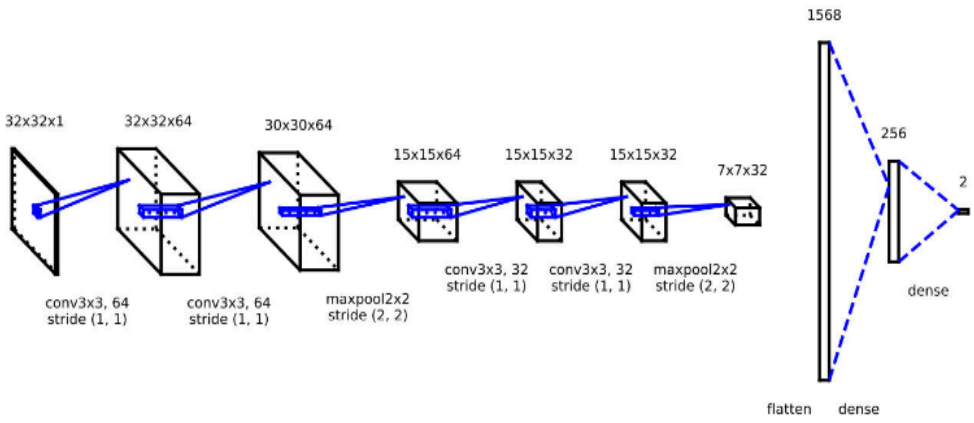


Fig: CNN model architecture for the institutional data set.

Result

Data Set	Model	Accuracy	False Alarm Rate
Institutional	Ensemble Learning with SVM	0.9768	0.0010
	Ensemble Learning with KNN	0.9931	0.0010
	Ensemble Learning with Naive Bayes	0.9806	0.0011
	Ensemble Learning with Logistic Regression	0.9790	0.0010
	CNN	0.9968	0.0008
UNSW-NB15	Ensemble Learning with SVM	0.9902	0.0051
	Ensemble Learning with KNN	0.9960	0.0051
	Ensemble Learning with Naive Bayes	0.9818	0.0049
	Ensemble Learning with Logistic Regression	0.9896	0.0037
	CNN	0.9885	0.0041

Learnings and Findings

Building on previous research, combining Convolutional Neural Networks (CNNs) with ensemble learning can significantly boost how well we detect unusual network activity. CNNs excel at picking out important patterns from complex data, like network packets. When these features are used in ensemble models, such as Random Forests, which combine the predictions of multiple models, it improves the accuracy and reduces false alarms. This approach means using CNNs to identify key patterns and then applying ensemble methods to make the final detection more reliable and precise.

Paper 4

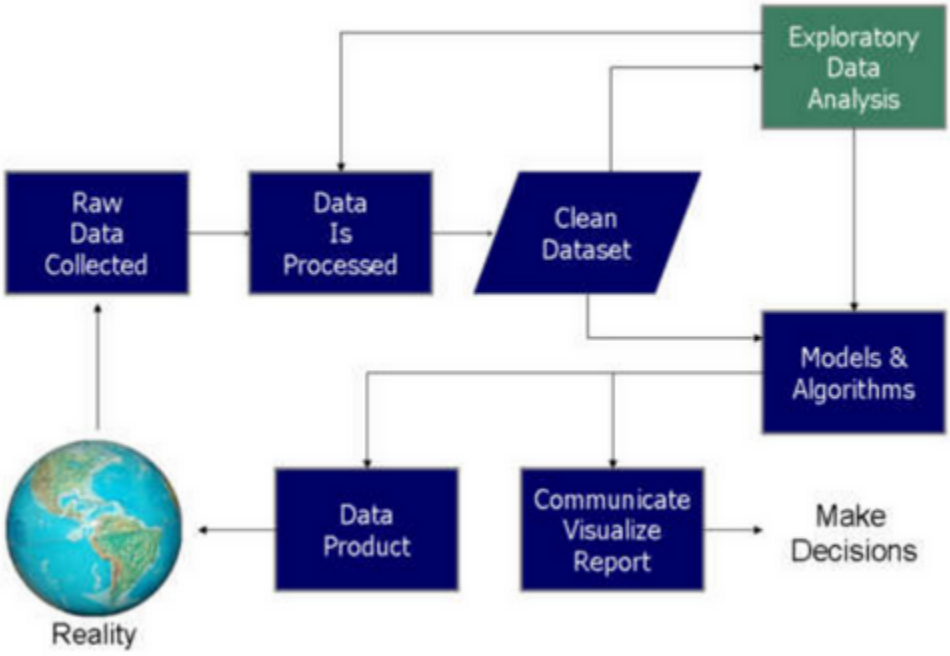
Title: Anomaly based Network Intrusion Detection using Data Science Process

Author: E. Vijay Kumar and D. Teja Santosh

The rapid growth of internet technology has increased the need for monitoring networks to prevent security issues. Networks are watched closely for unusual packets that might try to gain unauthorized access. To improve detection of these threats, data scientists use various techniques. First, they analyze the data to find the most important features that help identify attacks. Then, they use a method called t-SNE to reduce the complexity of the data while preserving its key features. Finally, they use these features in a random forest model, which has proven to be highly accurate, detecting 99.8% of anomalies and reducing false alarms. Deep learning methods like Convolutional Neural Networks (CNN) and Autoencoders also show high accuracy, but the random forest model with the data science approach offers the best results in terms of accuracy and fewer false positives.

Datasets used: NSL-KDD

PROCEDURE USED



1. **Datasets**
   The NSL-KDD dataset is used for analyzing network traffic and detecting attacks because it is a cleaned and balanced version of the older KDD99 dataset, with duplicate records removed. It contains 173,709 records and 43 features, including details about different types of attacks like Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). The dataset is imbalanced, with some attack types being much more common than others. For example, DoS and Probe attacks are much more frequent compared to U2R and R2L attacks. This imbalance is shown in boxplots, which reveal that U2R and R2L attacks are less common and less skewed compared to the more frequent DoS and Probe attacks.

2. **Machine Learning phase**
   a. Shallow Machine Learning using Random Forest Algorithm

      The Random Forest algorithm is chosen for training the classifier because it helps improve accuracy by reducing errors that might come from a single decision tree. It works by creating many decision trees and then combining their predictions. The final decision is based on the majority vote from these trees, making the algorithm more reliable and stable. During training, different settings or "hyper-parameters" are adjusted to fine-tune the model's performance, ensuring the classifier works as effectively as possible.

Machine Learning criterion	Value considered
Classifier learned	Random Forest
Number of predictors	32
Bootstrapping	True
Splitting Criterion	Gini
Maximum number of features allowed to try in individual estimator	Auto
Number of Estimators (Decision Tree as Estimator)	40

Fig:Hyper-parameters and Parameters for training Random Forest Classifier

b. Deep Learning using Autoencoder and CNN

Deep learning techniques are used for predicting future attacks because they can adapt to changing attack methods and handle new data more effectively than traditional methods. While previous models like RBM achieved 91% accuracy and XGBoost-DNN reached 97% accuracy on the NSL-KDD dataset, the proposed Autoencoder-to-CNN (A2CNN) model aims to improve this further. The A2CNN model first uses an autoencoder to learn and simplify feature representations from the dataset. After training, it discards the encoder's output and feeds the simplified features into a CNN for classification. The CNN then learns to distinguish between normal and anomalous packets using a binary-cross entropy loss function and Adam optimizer to refine its accuracy.

Results

The overview of NSL-KDD datasets are given below:

Table 4. Overview of the NSL-KDD dataset

Total	Normal	DoS	Probe	U2R	R2L
25192	13499	9234	2289	11	209
125973	67343	45927	11656	52	995
22544	9711	7458	2421	200	2654

The below curve shows the validation curve with Random Forest algorithm

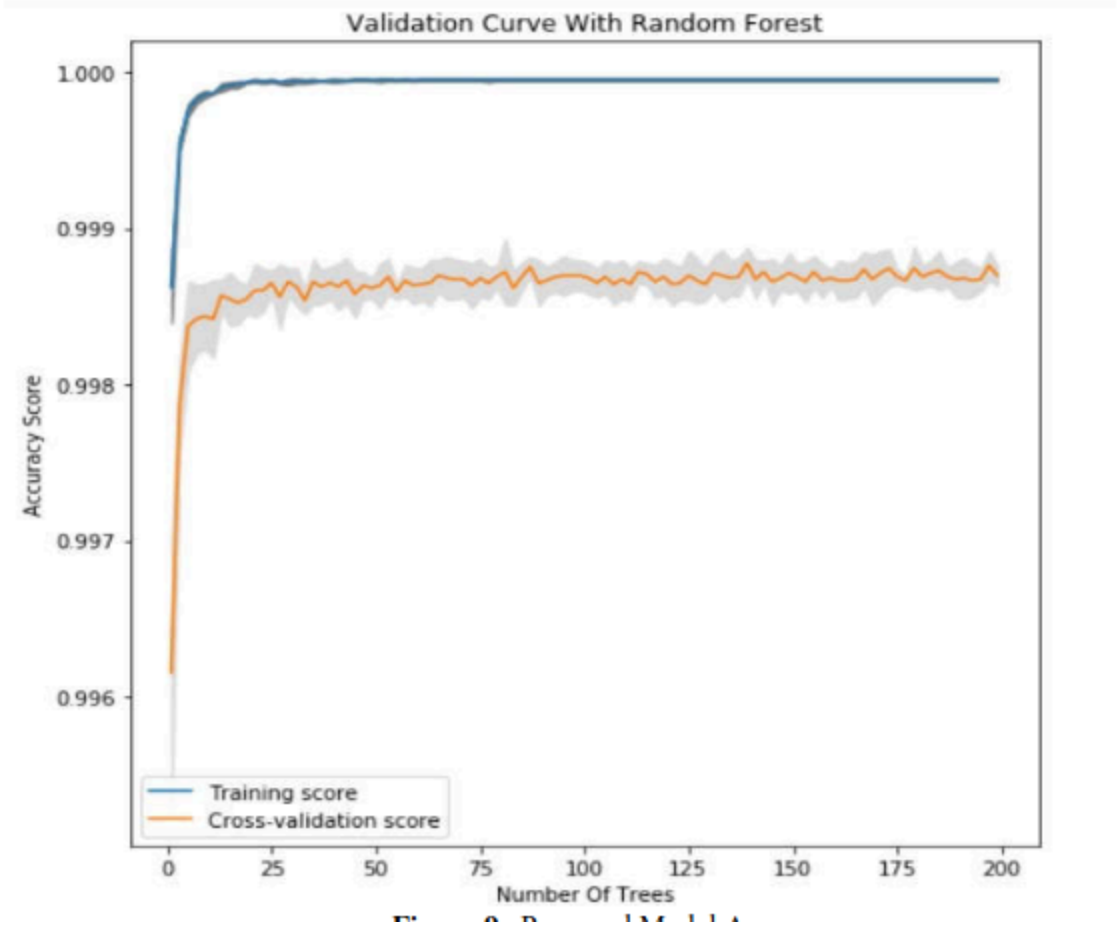


Fig: Proposed Model Accuracy

The performance metrics compared are namely Accuracy, Precision, Recall and False Alarm Rate. The comparison is on the basis of NSL-KDD dataset utilized in the state-of-art works.

Author	Technique used	Feature selection method	Accuracy (%)	Precision (%)	Recall (%)	False Alarm Rate
Tama et al., [4]	Adaboost and Random subspace model	Particle Swarm Optimization	97.5	97.3	98	0.03
Nandini and Manamohan [5]	k-means and Gradient Boosted Tree	k-means	99.6	99.3	-	0.19
Proposed Model	**t-SNE and Random Forest Algorithm**	**Chi-square Test**	**99.8**	**99.9**	**99.7**	**0.0006**
	Autoencoder and Random Forest Algorithm (Hybrid-1)		**99.4***	**99.5**	**99.2**	**0.003**
	t-SNE, Autoencoder and Random Forest Algorithm (Hybrid-2)		**99.5***	**99.6**	**99.4**	**0.002**

Fig: Performance Evaluation of the Random Forest Model with state-of-art models worked on NSL-KDD dataset.

The Hybrid-1 model achieves 99.4% accuracy because it uses autoencoders to focus on the overall variance of data, though they sometimes miss local details. The Hybrid-2 model improves on this with 99.5% accuracy by using t-SNE, which better retains local data structure. The proposed A2CNN model performs slightly better than methods using Particle Swarm Optimization (PSO) and k-means clustering, as it selects features more effectively with the chi-square test. This helps the A2CNN model achieve high accuracy and reliability in detecting network intrusions, with its performance metrics and results being among the best reported for this type of analysis.

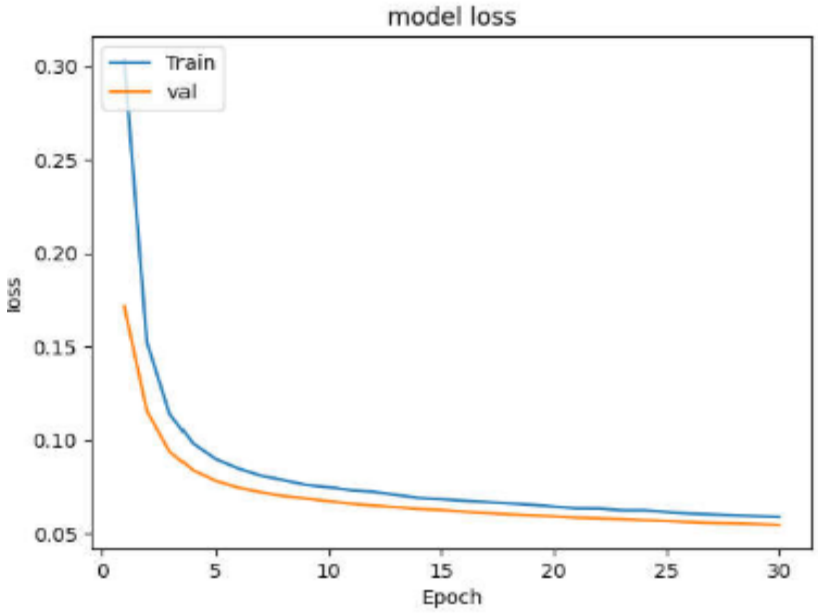


Fig: A2CNN model loss

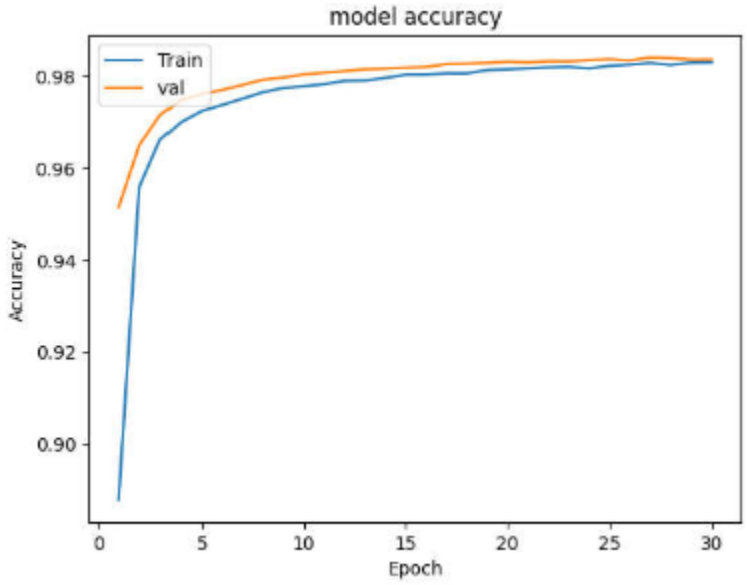


Fig: A2CNN model accuracy

The comparative analysis of the accuracies of the proposed A2CNN model with the state of the art deep learning models worked on NSL-KDD datasets for network intrusion detection are tabulated in below table.

Author	Technique used	Feature Selection method	Accuracy (%)
Yin et al., [6]	Recurrent Neural Network	-	83.28
Qureshi et al., [7]	deep sparse auto-encoder and self-taught learning	Regression related task	84.6
Tang et al., [8]	Autoencoder	LightGBM	89.82
Nguyen et al., [9]	PCA based Restricted Boltzmann Machine	-	91
Paul et al., [10]	Deep Neural Network	Correlation based feature selection	94.3
Preethi et al., [11]	Deep Neural Network	XGBoost	97
Proposed Work	**Autoencoder and CNN (A2CNN)**	**Chi-square test**	**98.4**

Literature Review Matrix

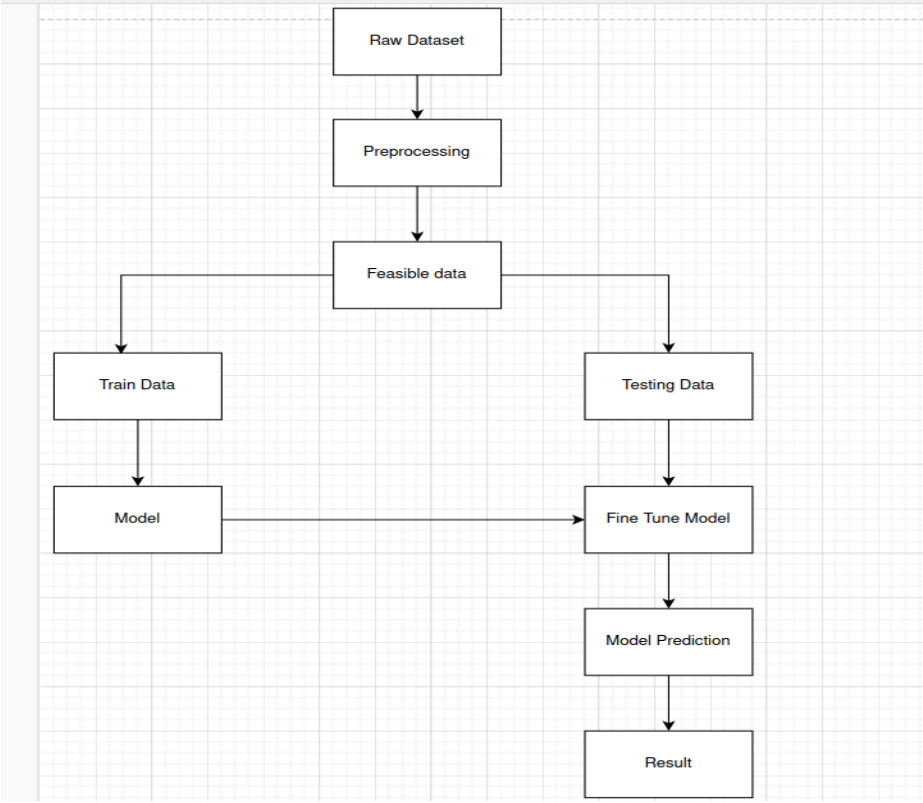
Title/Author/ Date	**Conceptual Framework**	**Research Question(s)/ Hypotheses**	**Datasets**	**Methodology**	**Analysis & Results**	**Conclusions**	**Implications for Future Research**
Intrusion Detection System Using Improved Convolution Neural Network - Xue Ying Li et al - 2022.	Using Improved CNN model, to enhance intrusion detection system performance.	Investigates if the improved CNN model can enhance accuracy and efficiency in IDS compared to traditional models.	NSL-KDD dataset, containing about 4,900,000 network connections	Improved CNN with Inception module, and performance evaluation.	Improved CNN model significantly outperforms the CNN-1 model	Improved CNN model effectively addresses traditional IDS challenges	Exploring distributed techniques for large-scale traffic management,
An intrusion detection system empowered by deep learning algorithms	Data preprocessing followed by image generation and applying deep learning frameworks	Analyzes the benefits of employing DL algorithms over ML algorithms for IDS	EdgeIIoT dataset	Using CNN based architectures like VisionTransformer, EfficientNetv2	H2O performed significantly better than others	Trained models show good result with realtime deployment possibility	Large scale network data collection and identifying new attacks

SOME/IP Intrusion Detection using Deep Learning-based Sequential Models in Automotive Ethernet Networks - Natasha Alkhatib, Hadi Ghauch, Jean-Luc Danger -2021	RNNs for intrusion detection in the SOME/IP protocol within automotive Ethernet networks	How can deep learning improve intrusion detection for the SOME/IP protocol?	Newly generated and publicly available dataset	Utilizes RNNs for offline intrusion detection, with data preprocessing and feature extraction from generated packet data.	RNN model achieves F1 Scores and AUC values greater than 0.8 for various intrusion types, demonstrating high accuracy in detecting in-vehicle intrusions.	RNN-based model effectively detects intrusions in SOME/IP.	exploration of deep learning models for real-time intrusion detection
Anomaly-Based Intrusion Detection Using Machine Learning: An Ensemble Approach- R. Lalduhsaka, Nilutpol Bora, Ajoy Kumar Khan-2022	Ensemble machine learning approach to improve detection accuracy and reduce false alarm rates.	How can an ensemble approach using machine learning improve the performance of anomaly-based IDS?	- CICIDS 2017: 2,830,743 records, 15 classes, 80 features - CICIDS 2018: 16,233,022 records, 18 classes, 83 features	Involves data preprocessing, feature selection with a random forest regressor, and an ensemble of Naïve Bayes, QDA, and ID3, evaluated against standalone models	- CICIDS 2017: Ensemble accuracy of 98.3%, false alarm rate (FAR) of 2% - CICIDS 2018: Ensemble accuracy of 95.1%, FAR of 6.9%	Ensemble approach significantly improves the performance of anomaly-based IDS,	Research on other machine learning algorithms and hybrid approaches is suggested

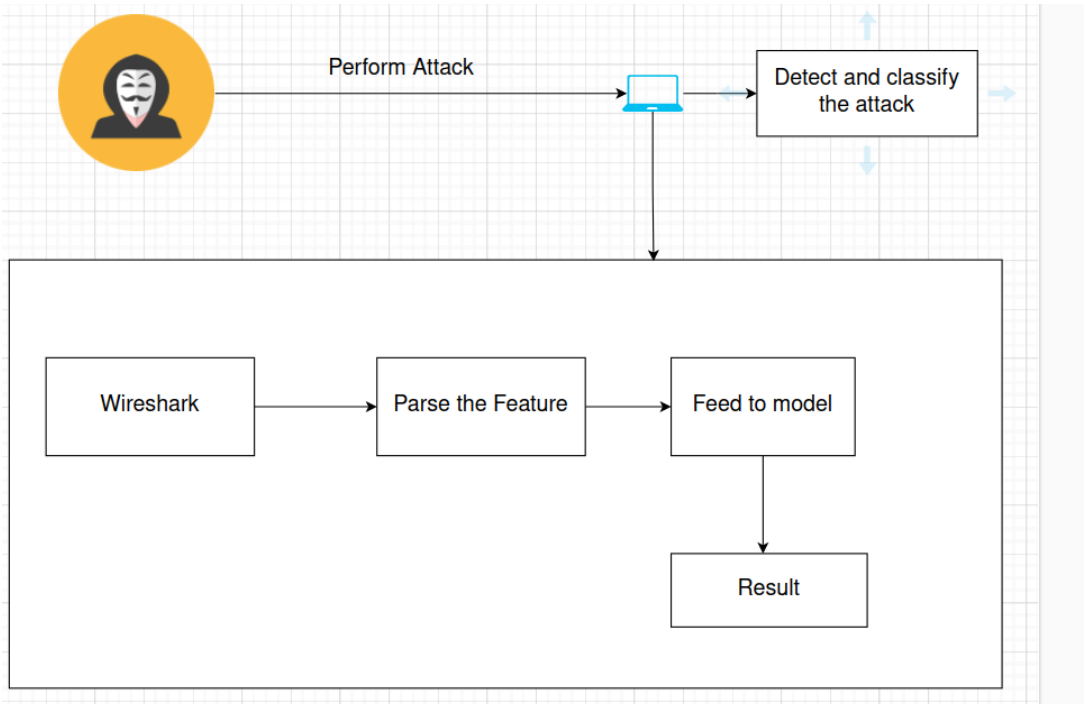
Anomaly based Network Intrusion Detection using Data Science Process - E. Vijay Kumar and D. Teja Santosh - 2020	Data science process applied to detect anomalous packets in network traffic	How does the IDS detect anomalous packets in the network to prevent future attacks?	NSL-KDD dataset, containing 173,709 records and 43 features	Random forest and deep learning model training	Random forest achieved 99.8% accuracy with a 0.0006 false alarm rate; Autoencoder-CNN model reached 98.4% accuracy	The integrated data science process effectively predicts network anomalies	Automated deep learning techniques for evolving attack patterns and unknown future threats

Project Solution Proposal:

Our data is mainly tabular so, we believe most of the effort will be utilized towards data engineering and analysis. However, we will train our models on both ML and DL algorithms to see if either outperforms the other. Also the time associated with each inference must also be considered if we are to implement this in real time. So, the project solution is shown in the figure below:



The dataset needs to be preprocessed so that many of the important features can be selected and properly scaled for creating a suitable representation for prediction models. The obtained data will be split and used to train and test the model.



In the end, we aim to create a system that can predict an attack or not in real time. To do so, we will use wireshark to obtain the necessary information from the packet and then analyze to determine its threat.

References:

1. Li, X. Y., et al. (2022). Intrusion Detection System Using Improved Convolution Neural Network.
2. Tang, W., Li, D., Fan, W., Liu, T., Chen, M., & Dib, O. (2023). An intrusion detection system empowered by deep learning algorithms.
3. Lalduhsaka, R., Bora, N., & Khan, A. K. (2022). Anomaly-Based Intrusion Detection Using Machine Learning: An Ensemble Approach.
4. Alkhatib, N., Ghauch, H., & Danger, J.-L. (n.d.). SOME/IP Intrusion Detection using Deep Learning-based Sequential Models in Automotive Ethernet Networks. LTCI, Telecom Paris, IP Paris, Palaiseau, France.
5. D. Teja Santosh Vijay Kumar. Anomaly-based network intrusion detection
using data science processes. XVI(IX):208–220, 2020.