National College of Ireland

Hybrid Anomaly Detection Framework for Kubernetes Environment

MSc Research Project
M.Sc. in Cybersecurity

Jai Allahabadi
Student ID: X23218193

School of Computing
National College of Ireland

Supervisor: Prof. Evgeniia Jayasekera

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Jai Allahabadi		
Student ID:	X23218193		
Programme:	M.Sc. in Cybersecurity	**Year:**	2024-25
Module:	M.Sc. Research Project		
Supervisor:	Evgeniia Jayasekera		
Submission Due Date:	12th December 2024		
Project Title:	Hybrid Anomaly Detection Framework for Kubernetes Environments		
Word Count:	6986	**Page Count:**	20

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: Jai Allahabadi

Date: 12th December 2024

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):	

Hybrid Anomaly Detection Framework for Kubernetes Environments

Jai Allahabadi
X23218193

Abstract

Over 60% of the enterprises have adopted Kubernetes and as per CNCF survey, the adoption rates have been increased to 96%. With such a high adoption rate, security concerns also arise exponentially. The market size for K8s security will be projected to reach $27.19 billion by 2032. Hence, the need to delve into the security of the K8s has become the need of the hour. With the advancement of artificial intelligence, the intrusion of the AI algorithms for anomaly detection has been significantly increasing. This paper builds upon the hybrid model that employs Long-Short Term Memory (LSTM), custom attention layer and Transformer network, for detection of anomalies along with the help of feature engineering techniques i.e., Principal Component Analysis (PCA) and Autoencoders. The hybrid model has been trained using traditional and Model-Agnostic Meta Learning (MAML) methods. NSL KDD and Kubernetes based attacks datasets have been employed in this research. Extensive experiments have been stemmed from an intent to explore the synergy between feature engineering techniques and training methods, with the conclusion that hybrid model trained on Autoencoder features data using traditional method surpasses with 98% accuracy and 0.98 F1 score. However, training the hybrid model trained using MAML reduces the training time up to 99% compared to traditional method.

Keywords – Kubernetes, Anomaly Detection, Principal Component Analysis (PCA), Autoencoder, LSTM, Attention Mechanisms, Transformer Model, MAML

1 Introduction

Intrusion Detection System (IDS) is one the critical security components of the infrastructure of the enterprise's system, protecting it against the threat actors, attacking on the system using variety of attacks. IDS is generally classified into Host-based IDS (HIDS) and Network-based IDS (NIDS). HIDS often have advantages over NIDS as it has access of encrypted information or data packets unlike NIDS. However, it is much more difficult to configure and manage over the enterprise's host. Hence, NIDS are often used in most enterprises to protect their system from outside or inside threats by monitoring and detecting their systems.

Recent efforts in IDS are collaborating with Machine Learning (ML) or Artificial Intelligence (AI), evolving the anomaly detection process from traditionally statistical modelling to advanced machine or deep learning algorithms. Originated by Dorothy E. Denning in 1980's by introducing foundational work in anomaly detection using statistical method (Anomaly Detection History, 2021) to new age hybrid deep learning anomaly detection model, indicating that anomaly detection also been changing dynamically with ever-evolving technology.

Signature-based detection mechanisms, where the anomaly or outlier will be characterized by the known patterns/behaviors of threats, which fail often against the zero-day or sophisticated attacks. However, with the intrusion of machine/deep learning algorithms into anomaly detection, can identify evolving threats, with the help of data-driven insights. This (Chalapathy & Chawla, 2019) paper comprehensively studied many deep learning algorithms

and focused on how deep learning algorithms can be used in different applications or domains or environments. In (Pang et al., 2020) as well, 12 diverse modelling perspectives have been extensively reviewed for anomaly detection and how deep learning algorithms can be harnessed.

This research will employ deep learning algorithms to identify anomalies based on several types of attacks on Kubernetes clusters, which have been recorded in K8s dataset. In this study, we propose a novel hybrid model to identify anomalies in the dataset, by incorporating advance feature engineering techniques along with deep learning algorithms and meta learning to identify sophisticated attacks. In this study, after several experiments, we have built hybrid model with combination of feature engineering techniques and training methods and compare them based on various evaluation metrics and context.

RQ – How can we optimize hybrid model for anomaly detection in Kubernetes environment by using efficient feature engineering technique and training method?

To achieve the objectives of this research, we will perform various experiments using different techniques and models, to achieve the novel anomaly detection model. We will compare two feature engineering models where we employ Principal Component Analysis (PCA) and Autoencoders and train our models on them, whereas our hybrid model, which employs deep learning algorithm, LSTM, along with custom attention layer and Transformer network trained with both traditional and meta-learning (MAML) method, making the anomaly detection model robust and applicable to real-world applications.

The major contribution of this research is :

1. **Design and implementation of hybrid model:** One of the main objective of this research is to design and implement hybrid anomaly detection model with efficient feature engineering and training methods that has been developed and trained on K8s clusters data, which can be deployed to k8s clusters to identify the anomaly in the system environment.
2. **Evaluation of model performance:** Model performance has been evaluated using a variety of evaluation metrics such as accuracy, which will highlight the how hybrid model has performed on the data along with its applicability and robustness.
3. **Model performance comparison:** Another goal of this research is to compare the model performance which will help us in supporting our hybrid model to determine its accuracy, robustness, and adaptability.

The subsequent sections of this paper has been organized as follows: Section 2 will provide detailed literature review which delves into the context and background notable works in this field, followed by research methodology (Section 3), which delves into the approaches that have been followed during the research. Following that will be the design specification and implementation (Section 4) showcasing the employed feature engineering techniques, models and training techniques, in this study, followed by evaluation (Section 5) which showcases evaluation of our model performances, concluding it, with the conclusion and future work (Section 6).

2 Related Work

In the recent years of exploration in the field of anomaly detection and Kubernetes, thorough research has been conducted. Some of the notable studies that have been done have been explored on which this research builds upon and interacts with.

2.1 Traditional Methods

Traditional methods such as Signature based Intrusion Detection System (SIDS) can detect known attacks based on predefined rules or signatures of known attacks. The demonstration of SIDS's has been done in (Díaz-Verdejo et al., 2022), where three SIDS (Snort, ModSecurity and Nemesida Free) have been used with changing the configuration rules experimentally to find optimal number of rules for efficient performance of these SIDS, whereas, in (Asad et al., 2024), two SIDS, Snort and Suricata, have been comparatively experimented on labelled PCAP data and evaluated each IDS performance. However, the major challenges that are faced by SIDS are first and foremost, is being not able to detect zero-day or sophisticated attacks and secondly is that high number of rules can impact false positives, therefore need to experiment and find optimal number of rules that can be configured for efficient performance of SIDS.

2.2 Inclusion of Machine Learning in Anomaly-based IDS

Due to these challenges, shift of the research have been delve into Anomaly-based IDS (AIDS), where machine learning and deep learning algorithms have been employed to detect the anomaly in the behavior of the data. In recent years, many researches have been conducted in this field, where (Musa et al., 2020) showcases several research papers (published between 2015 and 2020) and (Nassif et al., 2021) provides more comprehensive overview of machine learning journey in anomaly detection by reviewing more than 290 research papers over a period of 20 years till 2020. In (Nassif et al., 2021), authors conclude that 27% of papers have applied unsupervised, followed by 18% applied supervised, whereas 7% have applied hybrid approach. This (Musa et al., 2020) paper also second with the conclusion with ensemble or hybrid methods have achieved better results than single classifiers. In the (Araujo et al., 2023), authors also tried to fine tune the machine learning algorithms for microservice based application by employing Random Forest and Decision trees and track the model how they perform in pre attack, during attack and post attacks scenarios, whereas in (Yolchuyev, 2023), it goes further comparing selective machine and deep learning algorithms and showcases the comparison between them, with the conclusion as XG boost achieving highest accuracy of 98.86%. In this paper, author (Aly et al., 2024) has used the same dataset, with 91% accuracy, which has been created and used in (Tien et al., 2019), where (Tien et al., 2019) has achieved 96% by their own novel solution anomaly detection model, Kub-Anomaly. However, in this research paper, two datasets have been employed, where one (Sever & Dogan, 2023) is specifically focused on K8s services data, which is more than 3500000+ rows and other is NSL KDD dataset – widely used for anomaly detection. Despite resolving challenges that traditional possess, machine learning algorithms are not effective for new-day attacks and do not perform well with sophisticated attacks. Therefore, the proposed model of this research addresses those challenges by adapting the deep learning algorithms along with attention mechanism to

emphasize on important time steps rather than single data point, as anomaly cannot be evident from a single data point.

2.3 Advancement of Anomaly-based IDS with Hybrid Deep Learning

With the advancement of artificial intelligence or deep learning algorithms, dynamics of anomaly detection are also changing significantly by employing these algorithms for detection of anomaly in the system. In notable study (Silva et al., 2022), where authors have employed LSTM, and an autoencoder, for anomaly detection achieving 95% ,whereas in another insightful study (Almaraz-Rivera, 2023), where authors have compared three deep learning algorithms i.e. One-Class Support Vector Machine (OC-SVM), Isolation Forest and Autoencoders, concluded the model performances by showcasing their best hyperparameters, where Autoencoders are outperforming the other two algorithms, having AUC of 0.88. Both (Silva et al., 2022) and (Almaraz-Rivera, 2023) have employed deep learning algorithms but they lack the attention mechanisms in their model which will help in focusing on critical time steps and give relative importance with other data points making it less effective to sophisticated attacks, whereas the proposed model in this study have incorporated attention layer along with transformer network which makes our model more robust for real world applications.

In the insightful study, (Aly et al., 2024) proposed a solution where they employed Principal Component Analysis (PCA) and autoencoders with Naïve Bayes classifier to obtain F1 score of 0.95 and accuracy of 91%, having low accuracy, whereas, in (Kale et al., 2022), authors have proposed a solution by integrating K-means clustering, GANomaly and One-Class Support Vector Machine (OC-SVM) and employed three dataset to test their model, achieving up to 99% in TON_IOT dataset, whilst , 71% in CIC -IDS 2018, showing significant difference in their model performance, making this model unstable.

After the release of the paper “Attention is all you need,” authors (Vaswani et.al., 2017) have introduced transformer model using attention mechanisms. Authors (Wang et al., 2022), implemented unsupervised anomaly detection model based on variational transformer, to capture potential correlation between sequences in time series data and compare the model performance with several models such as GDN, LSTM-AE, Isolation Forest and others, demonstrating up to 94% accuracy of this model along with surpassing all the models, which shows that transformer model are able to understand the complex relationship over the series of data rather than short span of memory. Both research papers MAVAE (Moon et al., 2023) and TranAD (Tuli et al., 2022) have applied meta learning, where MAVAE have been built on variational autoencoders trained using MAML, whereas TranAD utilizes deep transformed network model trained with meta learning MAML for better adaptation, and showed their experimentation results of both model performance, achieving 17% better F1 score than compared models. While MAVAE and TranAD utilizes transformer network along with meta learning via MAML, are based on predefined assumptions and lack of feature reduction techniques, making their model more computationally exhaustive with high dimensioned data, whereas the framework proposed in this study have employed feature extraction techniques to highlight important data representations, making this model more robust for real-world applications. Moreover, the proposed framework focuses more on Kubernetes based attacks, whereas these research papers implemented respective model on generalized dataset.

Authors	Contribution	Key Findings	Strengths & Weakness
Tien et al., 2019	Designed a novel solution, Kub Anomaly, specifically for anomaly detection in Kubernetes platform.	Accuracy of 96% has been achieved with their validations.	Strengths: Trained and evaluated on real data which has been set up for four months and real attacks have been conducted by attackers. Weakness: Limited dataset having 40,000+ rows only.
Silva et al., 2022	Employed LSTM with autoencoder for detecting anomaly by collecting the data using eBPF from containers.	Through experimentation of hyperparameters, have achieved 1.16 false positives rate across clusters.	Strengths: Fine tuning of hyperparameters of LSTM with the autoencoders. Weakness: Limited discussion on diverse dataset.
Almaraz-Rivera, 2023	Fine tuning of three algorithms i.e., OC-SVM, Isolation Forest and Autoencoders, based on Kubernetes platform.	Efficient hyperparameter tuning of Auto-encoders having AUC score of 0.88	Strengths: Hyperparameter tuning with empirical validation. Weakness: Lack of innovatory solution.
Aly et al., 2024	Proposed novel solution by employing Principal Component Analysis (PCA) and Autoencoders along with Naïve Bayes classifier.	Implemented multi class anomaly detection model with accuracy of 91% and F1 score of 0.95	Strengths: Implemented multi-class detection model with promising results of accuracy 91%. Weakness: Limited dataset and lack of comparative analysis.
Wang et al., 2022	Implemented Variational Transformer on time series data and compare the performance with GDN, LSTM-AE, Isolation Forest.	Empirical Validation for comparison between the proposed model achieving 94% accuracy	Strengths: Integrated variational Autoencoders with transformers achieving accuracy of 94%. Weakness: High risk of overfitting due to absence of regularization or adversarial training.
Moon et al., 2023	Novel solution, MAVAE, have been proposed where variational autoencoders have been trained using meta learning (MAML) technique.	Achieved improvement in prediction by 45% compared to variational autoencoders.	Strengths: Using variational autoencoders with meta learning for better adaptability. Weakness: Limited Noise differentiation, due to lack of additional attention layer for reducing noise.
Tuli et al., 2022	Novel solution, TranAD, have been implemented, based on transformer network trained using meta learning (MAML) technique.	Impressive improvement on training times by up to 99%.	Strengths: Using transformer mechanisms with meta learning. Weakness: Limited Noise differentiation.

Table 1: Summary of Literature Review

3 Research Methodology

Research Methodology is a systematic process of research which explores areas of designing, conducting, analyzing, and reporting the research conducted. It helps us to have a clear and logical framework to give the research legitimacy and helps in providing scientifically sound findings. (Almaraz-Rivera, 2023), (Kosińska & Tobiasz, 2022) and several others research projects have employed Experimental Research Design (ERD) research methodology to accomplish their research objectives. In this research implementation as well, ERD has been employed which will ensure sound findings for the proposed model with practical relevance in real world K8s environments.

Experimental Research Design (ERD) is a logical framework to conduct research with experiments using a set of variables. Experiments are designed and conducted to test the hypothesis with sound findings. Key components in ERD that one needs to follow:

I. *Identify the variables* – Identify the independent and dependent variables.
II. *Define Your Hypothesis* – Testable hypotheses are defined which have expected outcome to the research question. In this project two hypothesis have been defined to experiment on, stated as follows:

Hypothesis 1: For Feature Engineering

H1– "Autoencoders are more effective feature reduction technique for anomaly detection in multivariate time-series data than Principal Component Analysis (PCA), achieving higher accuracy on training on hybrid model."

Hypothesis 2: Training Techniques

H2: "Traditional training techniques outperforms meta learning-based technique MAML for binary class anomaly detection in terms of accuracy, with ample training data."

III. *Design an Experiment* – Design how experiment should be conducted and what are the requirements, controlled variables such as dataset size.
IV. *Implementation* – Experiments have been conducted by manipulation of independent variables to form concrete dependent variables as an output. Example can be finetuning our hybrid models, or how hybrid model reacts for each feature engineering technique.
V. *Evaluation* – Most critical part of the research is to evaluate the performance of the proposed model with notable scientific finding. In this study, different evaluation metrics have been employed to measure the outcomes of the experiments performed.

To test these hypotheses, several experiments have been conducted to measure set of variables to provide sound findings on the hypotheses. In this research project as well, experiments were conducted to test these hypotheses. For hypothesis 1, PCA and autoencoders have been employed as a feature reduction technique, which further trained on hybrid model, which then evaluated each technique performance on basis of different evaluation metrics such

as accuracy. For hypothesis 2, both feature engineering techniques data have been trained using both traditional training technique and using meta learning-based techniques MAML and will be evaluated on the basis of different metrics.

4 Design Specification and Implementation

In this study, hybrid anomaly detection model has been designed for K8s environment to enhance the performance of earlier classification models. In this section, techniques, architecture, and implementation of the model will be discussed, starting with dataset description.

4.1 Dataset Description

With ever-expanding area of research in anomaly detection, many researchers have contributed to build specialized datasets, which aims to further deep dive into the K8s network security. In this research as well, we have employed (Sever & Dogan, 2023) dataset, which have been compiled by exploiting various vulnerabilities and perform attacks on K8s cluster and compiled dataset, having combination of 10 types of attacks varying the count from only 34 to 156614, encoding to binary label having class distribution as shown in Fig 1. By employing this dataset and developing anomaly detection model based on the understanding of the data will help us build a model which will perform efficiently for K8s.

Along with this dataset, NSL-KDD dataset has also been employed which is widely used for the research of anomaly detection for network security, where four types of attacks data have been labelled, which have been encoded to binary class with class contribution as shown in Figure 2. NSL KDD is a subset of KDD'99 data, where redundant data has been removed, which will help in training the model. It will also help in supporting the evaluation of our model's performance by comparing it with existing models.

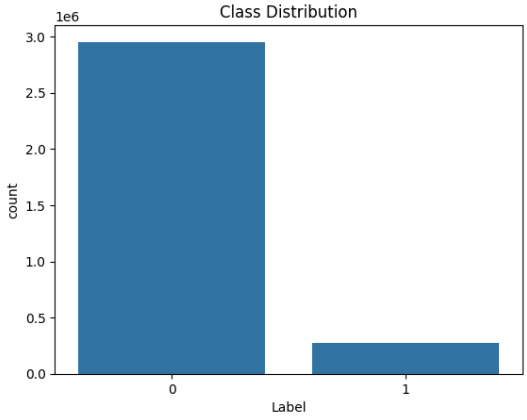

Figure 1: (Kubernetes) dataset Label count

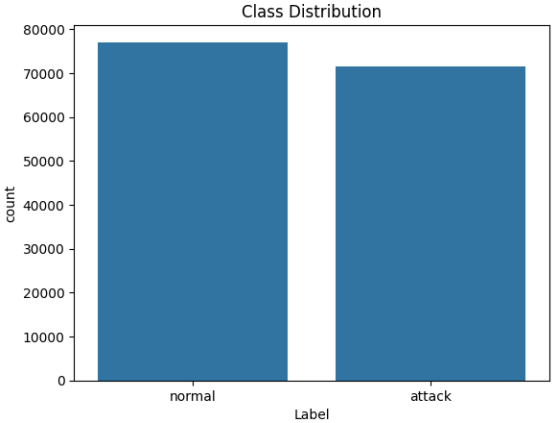

Figure 2: NSL KDD Label Count

4.2 Data Preprocessing

In data preprocessing stage, data is loaded and processed in such a way that useful information can be used further for training the model. From cleaning the data which includes, handling missing values, removing duplicates, and likewise processing has been done on the data. Furthermore, exploratory data analysis is also being done to understand the

dependent ('label') and independent (all other features of dataset) values, by understanding the distribution of data, correlations between features and dependent variables.

In (Sever & Dogan, 2023) dataset, pcap flow data has been captured which includes many important features such as protocol – defines the protocol used for communication, Flow duration – total time of the flow in microseconds, total FWD packets and total BWD packets – defines total number of packets flows between source and destination and it length are defined by - total FWD packets length, total BWD packets length. Other features in this dataset which might be important for dependent variable includes flow IAT mean, flow IAT std, flow packets/s, Idle mean, FIN/SYN/URG/CWR flag counts (individual features). Then, total TCP Flow Time which defines the total time duration of TCP flows and finally the dependent variable label, which defines whether it is benign or anomaly. Different network attacks or exploitation of vulnerabilities have been performed. As this dataset has ten distinct types of attacks varying the count from only 34 to 156614, which has significant difference, all anomalies have been labelled into one column through label encoding. Moreover, label encoding will also take place to convert other categorical values to numerical values using different techniques such as One-Hot encoding. Besides that, data transformation also takes place to normalize the data, by applying scaling techniques like StandardScaler or Min-Max scaling, which transforms the features in a specific range. StandardScaler technique, which has been used in this project for normalizing the data.

$$z = \frac{x_i - \mu}{\sigma}$$

Figure 3: Formula for StandardScaler

Where:

- z is the scaled value
- x_i is the feature value (original)
- μ defined as mean of the feature
- σ denotes to the standard deviation of the features.

Data balancing is part of data preprocessing where if dataset has imbalance data, where one label has significantly more values than other, then balancing of data should takes place, which will help in training the model to understand the complex relationship of independent and dependent variables. In this project, SMOTE (Synthetic Minority Oversampling Techniques) has been employed for oversampling of minority data (i.e. 'Anomaly' label) in Kubernetes dataset, which creates synthetic values by selecting the minority values at random and through k-nearest neighbors' algorithm, random values are selected which is closely related to the random values selected.

4.3 Dimensionality Reduction Techniques

Dimensionality reduction is a process of simplifying the data by reducing the number of features, preventing the effect of curse of dimensionality. It is very crucial to eliminate less-important features from the dataset to enhance the model performance. Several studies such as (Aly et al., 2024), (Zheng & Rakovski, 2021) and others have employed PCA and autoencoders for reducing features from the dataset. In this research, as ERD approach was employed, experiments have been carried using two dimensionality reduction techniques: Principal Component Analysis (PCA) and Autoencoders.

4.3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is dimensionality reduction technique which reduces the number of features from the dataset by understanding the importance of each feature based on their relationships and quantify that relationship using mathematical functions. This (Sever & Dogan, 2023) dataset have eighty-seven features including the dependent variable, and if model is trained on these many features, it might be possible that model performance can be degraded because of curse of dimensionality. Therefore, this dataset has been reshaped into combination of smaller sets, based on the uncorrelated components, also called as principal components, which converts higher dimensional space to combinations of lower dimensional space, as shown in below figure 4. Firstly, the features have been extracted out along with the dependent variable from the dataset and then applied PCA. Empirical studies such as (Hasan & Abdulazeez, 2021), shows that efficient variance for PCA is 95%, therefore, while applying PCA on this dataset, data variance of 95% has been set, which means that the dimensionality has been reduced while retaining 95% of total information or patterns of the original dataset. After applying PCA, the dataset has been reshaped to a new feature space basis on the principal component, which has been stored in data frame along with dependent (‘label’) variable.

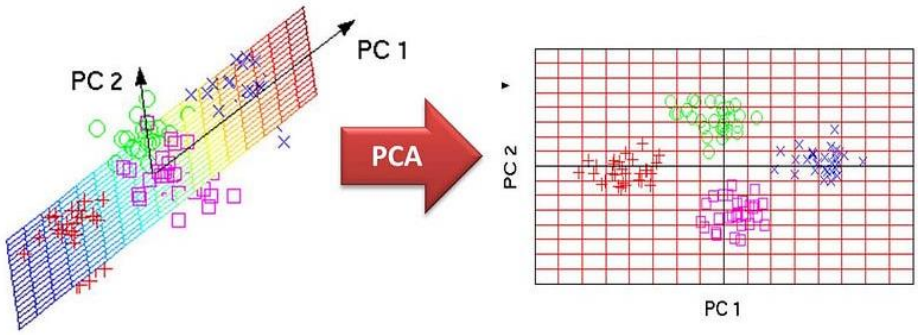

Figure 4: Principal Components

4.3.2 Autoencoders

Autoencoders are type of unsupervised neural network which is widely used for feature reductions or learning. It consists of encoder and decoder, where encoder compressed the input data into lower dimensional form, resulting smaller representations, whereas decoder decompressed the compressed data back to reconstruct to original input with minimal reconstruction loss, as shown in below figure 5. Both encoders and decoders use multiple neural network layers to perform their actions. While implementing it for feature reduction, the input layer has been fed with all the features of original dataset. Then, encoder layer will encode or compress the number of features into lower dimensions. The extracted features have been set off as ten, which is like PCA, and have been taken to compare both techniques. Once the encoder captures the valuable information, the decoding layer reconstructs it back to original form, with minimal reconstructing loss. Then the autoencoder was trained, with the help of Adam optimizer for efficient learning rate adaptation and mean squared error loss for minimal reconstruction loss. Once the model has been trained, then the encoder part of the model has been used to extract the transformed data, which is a compressed dataset features having lower dimensions. That extracted data from the encoder part of the model will be stored in a data frame along with dependent variable as reduced feature set, which will be used for training the hybrid model for anomaly detection.

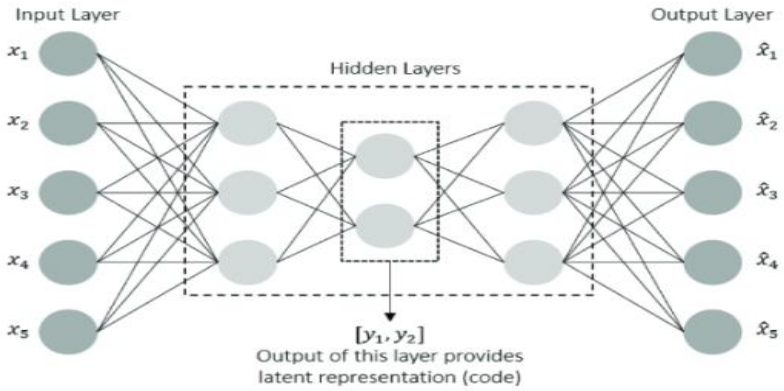

Figure 5 : Autoencoder Layers

4.4 Hybrid Model

In this experimental research, employed models namely LSTM and Transformer network along with custom attention layer, integrating it as hybrid model for anomaly detection designed for Kubernetes environments.

4.4.1 Long-Short Term Memory (LSTM)

Long-Short Term Memory (LSTM) is type of Recurrent Neural Network (RNN) that was proposed by Hochreiter and Schmidhuber in 1997 for understanding the short-term and long-term dependencies in sequential data. It addresses the vanishing gradient problem during the back propagation, as in the traditional RNN's, by adapting the new architectural design which introduces memory cells with gating mechanisms. Each LSTM block, as shown in fig, comprises of input gate, memory cell forget gate, output gate and candidate gate. (Song et al., 2020). In this research, LSTM has been employed as the first layer (after the input layer) to process the sequential data, where hidden state captures the information of previous steps. LSTM is extremely useful for capturing short-term and long-term dependencies over the data. In the case of anomaly detection, it becomes essential to capture both short-term and long-term dependencies to understand the data, which will be helpful to detect an anomaly, as anomaly may not be evident from single data point. Hence, layer of LSTM has been chosen with sixty-four units, to capture both dependencies and further output to subsequent layers for further operations. The input layer with all the features that have been extracted using feature engineering techniques have been fed, which process the features to make compatible for LSTM layer, where sixty-four units (memory cells) have been deployed, which is chosen for optimal tradeoffs between model capacity and computational efficiency. for capturing temporal dependencies over the sequential data, which further analyzed by the custom attention layer and transformer layer.

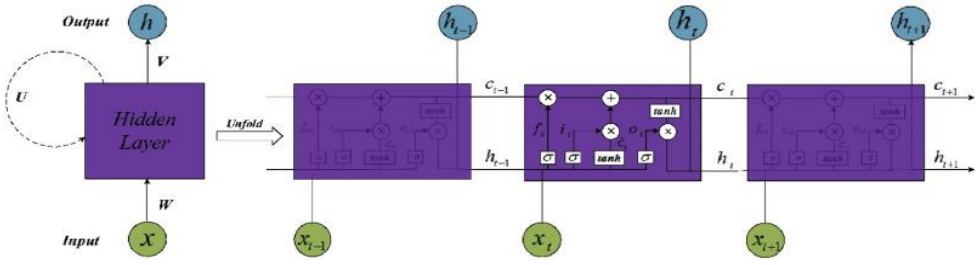

Figure 6: LSTM Layers

4.4.2 Custom Attention Layer

Once LSTM has processed the data to capture temporal dependencies, a custom attention layer has been added to focus more on relevant parts of the LSTM's output, with the help of trainable weights and biases. These trainable parameters give the importance or attention to each time step, which has been normalized using Softmax function, which helps in quantifying the relative importance of one-time step over the other. Using these trainable attention scores, context vector has been calculated, which is the sum of weights for LSTM outputs, where weights are calculated by the attention scores. The context vector will give the importance to the key time steps in the data. This custom attention layer will emphasize crucial time steps by adjusts the focus dynamically over the LSTM outputs, which will help the model in deeper understanding of the data.

4.4.3 Transformer

Transformer model, proposed by (Vaswani et al., 2017) in the seminal paper "Attention Is All You Need", whose architecture is constructed based on attention mechanisms, dispensing with recurrence and convolutions with multi-head attention layers. The architecture of transformer, as shown in below figure 7 as well, consists of encoder and decoder block, consisting of key components are self-attention layer or multi-head self attention layer – which is responsible for model's ability to focus more on relevant parts in the sequence data, followed by feedforward neural network – which provides fully connected neural networks for processing multiple output from multi-head attention layer to concatenate the attention outputs, having normalization layer in between them. The other advantage of employing transformer model is parallelization, as it processes the data parallelly, making the model faster and suitable for real time monitoring as well. Several experiments were conducted employing transformer models for anomaly detection by (Xu et al., 2022), (Tuli et al., 2022) and others, resulting in better performance of models in detecting anomalies due to its multi-head attention mechanisms along with feed forward network.

In this implementation, transformer has been applied for refining the features, The sequential and context aware data has been fed into the transformer model, where it has been input to multi-head attention layers, to capture each time step importance relative to relationship over all the data. (Vaswani et al., 2017), (Tuli et al., 2022) and several others have chosen 4 multi-head attention for balancing computational cost and efficiency. Hence, multi-head attention layer has been applied with 4 as number of attention heads for self attention mechanisms, where each head will capture the relationships between features, in addition with extended-attention output for context aware output, which further been passed to feed-forward network (ffn). Feed-forward network is densely neural network used for feature refinement. The dimensionality of the transformer model has been set to 64. With these configurations, transformer model has been tailored made for processing sequential network logs, which will capture global relationships or dependencies between features, improving the detection of anomalies, making the proposed hybrid model, robust and efficient.

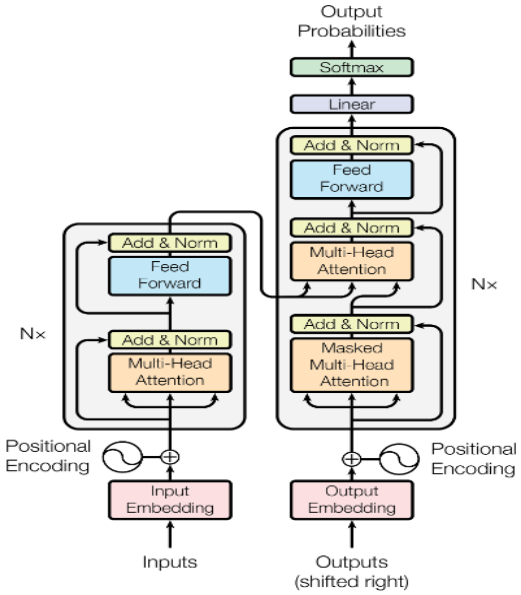

Figure 7: Transformer Model

4.5 Meta Learning Technique – MAML

Meta Learning Technique is new-age learning techniques for models where goal is to make the models adapt to unseen data with minimal training. In this learning technique, tasks are developed designed for adapting new data, and model will be trained on that task, with minimal training data. Unlike traditional method, meta learning focuses on learning for a specific defined tasks rather than general learning across data, which can help in training our model with minimal new data. Model-Agnostic Meta-Learning (MAML), is also type of meta learning technique proposed in paper "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks" by (Finn et al., 2017), with the purpose to make the meta learning model agnostic, which means that MAML is compatible with any model trained with gradient descent and can be applicable to variety of algorithms including classification, regression and reinforcement learning. It offers various advantages over traditional training methods, as it can detect unseen anomalies with minimal training of the model. Experimental results in (S. Tuli et. al, 2022), (J. Moon et. al., 2023) and other papers shown that by creating tasks with minimal training data can help the model in adapting to new tasks and performance are enhanced with improved accuracy and reduce false positive. In this research study as well, hybrid model has been trained with both traditional method and with MAML by creating set of tasks for distinguishing between normal and anomaly behavior, each tasks having small number of support set (for training) and query set (for testing). To train the hybrid model using MAML, two loops have been created, inner and outer loop where inner loop is responsible for task specific training whereas outer loop is responsible for meta parameters optimization. In the inner loop, gradient descent has been calculated using tasks' support set which will help in adjusting model parameters such as model's weights to minimize the loss for that task. Whereas, in outer loop, model's parameters have been set back to their original state, loss has been calculated for the query set, which then were used to optimize the initial model's parameters across all tasks. (Finn et al., 2017) shows that the most stable and best suited inner loop learning rate and meta learning rate are 0.01 and 0.001 respectively to avoid overfitting. Hence, the parameters have been set with these values for training the hybrid model. This process of meta learning repeats for multiple epochs to reduce the meta loss across all the tasks. Adam Optimizer has been employed for meta-gradient updates because of its adaptive learning rate capabilities.

4.6 Architecture of Anomaly Detection Framework

The design specification for this research with the proposed workflow with experiments conducted using the above explained feature engineering techniques, hybrid model and training techniques. Illustrated workflow, as shown in figure 8, shows the journey of data from loading the data to experimental steps that have been conducted throughout the workflow to final step prediction of anomaly. Below figure 8, have been designed to streamline the process for both, NSL KDD and Kubernetes anomaly datasets, for detecting whether the data belongs to anomaly or benign. The hybrid model that was proposed consists of LSTM, custom attention layer and transformer layer, which have been tailored made and applied with both traditional learning and meta-learning techniques to determine the anomaly efficiently along with its adaptability for Kubernetes environments. The implementation of proposed workflow has been in the following section.

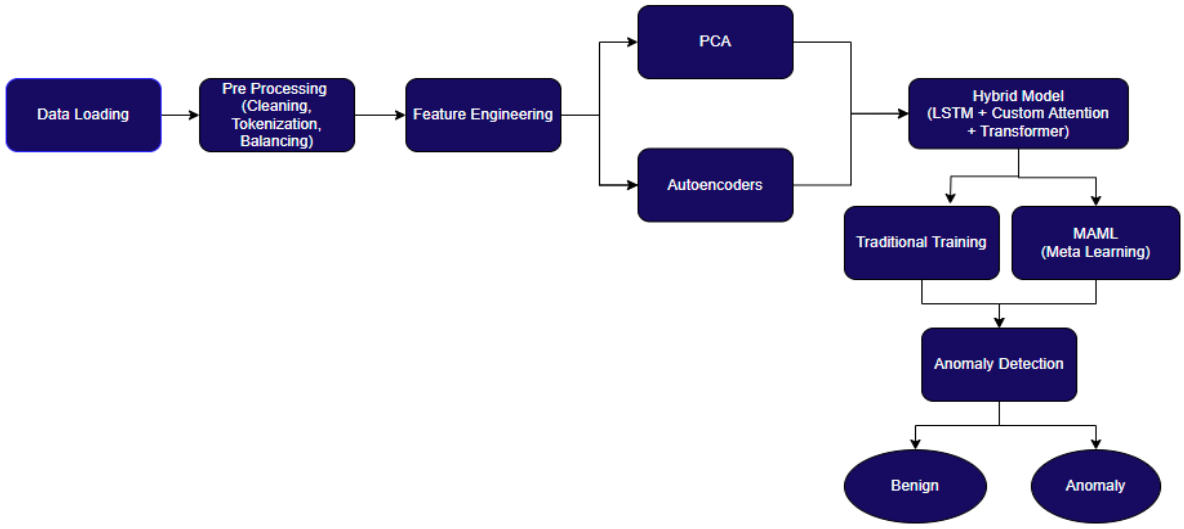

Figure 8: Proposed Workflow

4.7 Model Implementation

As this research builds through experiments of different techniques of feature engineering and training methods, distinct implementation of the hybrid model, with hybrid model summary illustrated in figure 9, have been explored. The workflow is same for all the experiments till data preprocessing stage where data have been processed for training the model. Each experiment uses different facets of techniques with the hybrid model as follows:

Hybrid Model = LSTM + Custom Attention Layer + Transformer

Hybrid Model over PCA data with Traditional training: In this experiment, after data have been pre-processed, feature engineering has been conducted using PCA for dimensionality reduction and fed the extracted data to the hybrid model and trained using the traditional training method using fit() method, aiming to examine the efficiency of PCA components from the processed data with traditional training method.

Hybrid Model over Autoencoders data with Traditional training: Here, the hybrid model has been fed on extracted data from autoencoders which have been used for dimensionality reduction and then the hybrid model has been trained using traditional method, aiming to

gauge the hybrid model performance on Autoencoder dimensionality reduced data by training with traditional method.

Hybrid Model over PCA data with MAML: The above two experiments, hybrid model was trained using the traditional training method, but in this variant, dimension reduced using PCA have been fed to the hybrid model which is trained using MAML technique by creating tasks for normal behaviour of data and hybrid model have been trained on the benign tasks, to examine the hybrid model ability to predict the benign or anomaly behaviour based on the tasks defined.

Hybrid Model over Autoencoder data with MAML: In this variant, Autoencoder have been employed for dimensionality reduction and then hybrid model has been applied to post-Autoencoder data through MAML technique by creating tasks for benign behaviour and evaluate the cumulative impact of Autoencoder-based dimensionality extraction on the hybrid model that have been trained using MAML learning technique.

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 1, 10)	0	-
lstm (LSTM)	(None, 1, 64)	19,200	input_layer[0][0]
attention (Attention)	(None, 64)	65	lstm[0][0]
expand_dims_layer (ExpandDimsLayer)	(None, 1, 64)	0	attention[0][0]
multi_head_attention (MultiHeadAttention)	(None, 1, 64)	66,368	expand_dims_layer[0][… expand_dims_layer[0][…
add (Add)	(None, 1, 64)	0	expand_dims_layer[0][… multi_head_attention[…
layer_normalization (LayerNormalization)	(None, 1, 64)	128	add[0][0]
dense (Dense)	(None, 1, 128)	8,320	layer_normalization[0…
dense_1 (Dense)	(None, 1, 64)	8,256	dense[0][0]
add_1 (Add)	(None, 1, 64)	0	layer_normalization[0… dense_1[0][0]
layer_normalization_1 (LayerNormalization)	(None, 1, 64)	128	add_1[0][0]
flatten (Flatten)	(None, 64)	0	layer_normalization_1…
dense_2 (Dense)	(None, 1)	65	flatten[0][0]

Figure 9: Hybrid Model Summary

All these experiments have been stemmed from an intent to explore the synergy between feature engineering techniques and training methods and being evaluated by employing variety of evaluation metrics and compare the results to form the conclusion for robust and efficient anomaly detection model framework.

5 Evaluation

Evaluation is most critical in Experimental Research Design methodology as it gives the testament for the hypothesis with sound findings. In this section, four experiments that have been done in this research will be evaluated and compared the results along with concluding two hypotheses that have been developed. Evaluation for each experiment have been done on the (Sever & Dogan, 2023) dataset explained and illustrated below. The best performing experiment will be employed for NSL KDD dataset. From section 5.1 to 5.4 have showcase

the evaluation for each demonstration on (Sever & Dogan, 2023) dataset and section 5.5 delves into the discussion of the best performing demonstration along with its results on NSL KDD dataset.

5.1 Exp 1 - Hybrid Model over PCA data with Traditional training

In this experiment, data extracted using PCA feature engineering technique have been fed to the hybrid model and trained using the traditional method. The classification report of this experiment has been shown below figure 10, which shows that it achieves accuracy of 95% with an average F1 score of 0.95. Also, tensor-board have been employed for capturing logs during the training of the model, and one of the important metrics (accuracy of epochs during training) have been shown below in figure 11.

	precision	recall	f1-score	support
Class Benign	0.96	0.94	0.95	442994
Class Anomaly	0.94	0.96	0.95	442994
accuracy			0.95	885988
macro avg	0.95	0.95	0.95	885988
weighted avg	0.95	0.95	0.95	885988

Figure 10: Classification report (Experiment 1)

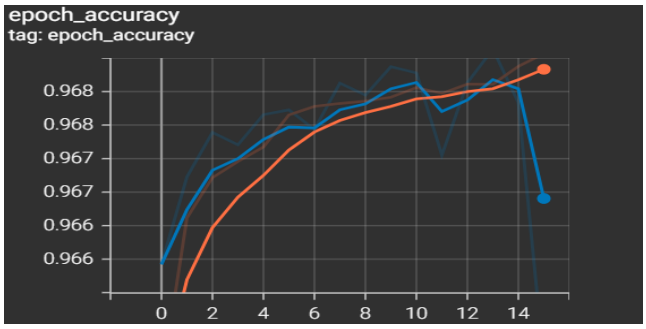

Figure 11: Epoch Accuracy during training (Experiment 1)

5.2 Exp 2 - Hybrid Model over PCA data with MAML

In this variant, hybrid models have been trained using MAML (meta-learning) technique on the data which have been dimensionality reduced using PCA technique. Accuracy and average F1 score are 92% and 0.92 respectively as shown in the classification report. By utilizing tensor-board logs, meta loss during training has been captured as shown below in figure 13.

Classification Report on Full Test Set:

	precision	recall	f1-score	support
Normal	0.95	0.90	0.92	442994
Anomalous	0.90	0.95	0.92	442994
accuracy			0.92	885988
macro avg	0.92	0.92	0.92	885988
weighted avg	0.92	0.92	0.92	885988

Figure 12: Classification Report (Experiment 2)

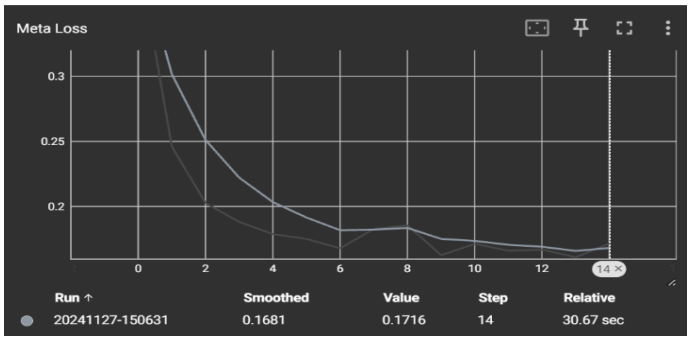

Figure 13: Meta Loss during training (Experiment 2)

5.3 Exp 3 - Hybrid Model over Autoencoder data with MAML

In exploration, where data have been dimensionally reduced using Autoencoders have been applied to the hybrid model which have been trained using MAML learning technique. This variant has achieved 94% and 0.94 for accuracy and average F1 score, respectively. Also, tensor-board have been employed to showcase the meta loss during the training of the model, as shown in figure 15.

Classification Report on Full Test Set:

	precision	recall	f1-score	support
Normal	0.95	0.92	0.94	442994
Anomalous	0.93	0.95	0.94	442994
accuracy			0.94	885988
macro avg	0.94	0.94	0.94	885988
weighted avg	0.94	0.94	0.94	885988

Figure 14: Classification report (Experiment 3)

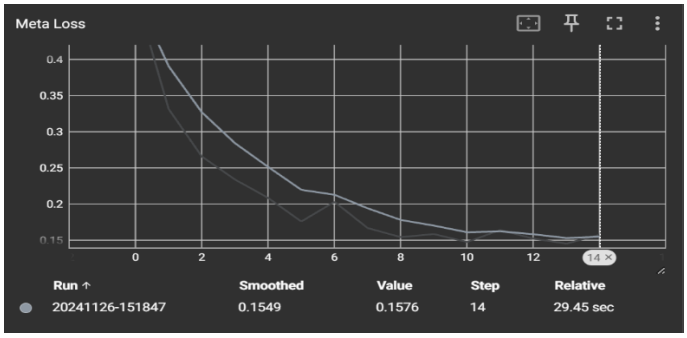

Figure 15: Meta Loss during training (Experiment 3)

5.4 Exp 4- Hybrid Model over Autoencoders data with Traditional training

In the last experiment, data that have been extracted using Autoencoders feature engineering technique have been applied to the hybrid model which have been trained with traditional method. As shown in the classification report, accuracy and average F1 score that have been achieved with this experiment are 97% and 0.97 respectively, along with epoch accuracy while training the model as shown in figure 17, captured by tensor-board logs.

	precision	recall	f1-score	support
Class Benign	0.99	0.94	0.97	442994
Class Anomaly	0.95	0.99	0.97	442994
accuracy			0.97	885988
macro avg	0.97	0.97	0.97	885988
weighted avg	0.97	0.97	0.97	885988

Figure 16: Classification Report (Experiment 4)

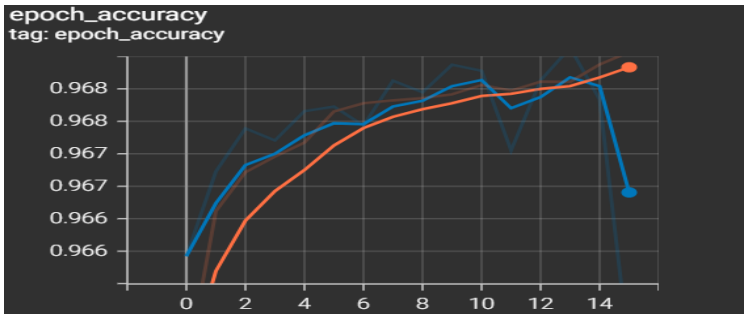

Figure 17: Epoch Accuracy during training (Experiment 4)

5.5 Discussion

After performing the different experiments on (Sever & Dogan, 2023) dataset and evaluating the model performance for each demonstration, it have been concluded that, employing Autoencoders for dimensionality reduction technique that have been fed to the hybrid model (LSTM, custom attention layer & Transformer) which have been trained using the traditional method have surpasses the other experiments, by achieving 97% and 0.97 accuracy and average F1 score respectively. Comparison of each demonstration have been shown in below table.

Feature Engineering	Training Methods	Accuracy	Training Time (in minutes)
PCA	MAML	92%	1.7
PCA	Traditional	95%	83.3
Autoencoders	MAML	94%	1.6
Autoencoders	**Traditional**	**97%**	**136.6**

Table 2: Comparison of hybrid model on Kubernetes dataset

The above results also give the testament with sound findings for both hypotheses that have been developed for this study. Hence, the hybrid model with Autoencoders trained with both traditional and MAML methods have been performed on NSL KDD dataset as well, where our hybrid framework has shown below results, as shown in figures.

Classification Report on NSD KDD Test Set:

	precision	recall	f1-score	support
Normal	0.90	0.98	0.93	15450
Anomalous	0.97	0.88	0.92	14254
accuracy			0.93	29704
macro avg	0.93	0.93	0.93	29704
weighted avg	0.93	0.93	0.93	29704

Figure 18: Classification report with MAML (NSL KDD dataset)

Classification Report on Test Set:

	precision	recall	f1-score	support
Normal	0.99	0.99	0.99	15450
Anomalous	0.98	0.98	0.98	14254
accuracy			0.98	29704
macro avg	0.98	0.98	0.98	29704
weighted avg	0.98	0.98	0.98	29704

Figure 19: Classification report with Traditional training (NSL KDD dataset)

NSL-KDD is widely used dataset for many researches in anomaly detection where (Zakariah et al., 2023) have reached accuracy of up to 97% and 0.90 F1 score whereas (Kale et al., 2022) have reached accuracy of 96% and 99 % respectively, but in both the papers, it failed with larger datasets, showing that their model is not robust for bigger size dataset. However, the proposed model in this study have showcase robust results with accuracy up to 98.5% and 0.98 F1 Score and with (Sever & Dogan, 2023) -larger dataset (having 3500000+ rows) achieving accuracy of 97% with 0.97 F1 score, making this model more robust and efficient.

6 Conclusion and Future Work

In this study, several experiments were demonstrated to build an efficient anomaly detection framework for Kubernetes environment, addressing the challenges in detecting anomaly in distributed systems. (Sever & Dogan, 2023) dataset based on the Kubernetes attacks where permutations and combinations with feature engineering techniques and training methods were conducted. Integrating the LSTM, custom self-attention layer and Transformer-based layer have been designed for building the hybrid model. Principal Component Analysis (PCA) and Autoencoders have been employed to reduce the features of datasets, which will be fed to the hybrid model, which will be trained with both traditional method and with meta-learning method via MAML. Each combination of feature reduction technique and training methods have been evaluated. The hybrid model with Autoencoder based dimensionality reduced method by training with traditional method surpasses other variants with 97% accuracy in Kubernetes based dataset and 98.5% accuracy in NSL KDD dataset. However, while training with MAML, the training time has been reduced to 99%. With this experimental setup, the objectives of this research have been achieved by testing two hypotheses that were proposed in the research. This anomaly detection framework will be served as valuable for future research and development in this field, along with addressing the practical aspect with deploying in real time.

This experimental based research successfully addresses the gaps that have been identified in the literature review along with successfully demonstrated effective anomaly detection framework making it suitable for real-world applications due to its high accuracy and efficiency. However, several challenges have been faced during the study, specifically integrating the hybrid model with LSTM and transformer networks to avoid overfitting and enhancing efficiency. Moreover, this research focuses on binary classification due to the low amount of data for each attack. Therefore, the multi-class large dataset can be made to further explore the multi-threat detection with this proposed framework.

Despite this success, several aspects can be explored in future research and development. In addition of how advance deep learning algorithms have been explored with feature engineering techniques, it can be further explored for multi-class threat detection with MAML

for multiple attacks on large datasets to understand subtle nuances of multi-vector attacks in Kubernetes cluster, by creating tasks for each attack class and train with MAML to deeply research in meta learning. Moreover, the anomaly detection model which has been demonstrated in this study can be implemented in Zero Trust Network architecture to enhance the security posture of the Kubernetes clusters. Therefore, by making continuous efforts to enrich the dataset with new-age attacks and patterns, further advancements in the field of IDS can be explored with ever-changing cybersecurity landscape.

7 References

Almaraz-Rivera, J. G. (2023). An anomaly-based detection system for monitoring kubernetes infrastructures. *IEEE Latin America Transactions*, 457-465. Retrieved from https://ieeexplore.ieee.org/abstract/document/10068850

Aly, A., Fayez, M., Al-Qutt, M., & M. Hamad, A. (2024). Multi-class threat detection using neural network and machine learning approaches in kubernetes environments. *2024 6th International Conference on Computing and Informatics (ICCI)*, (pp. 103-108). doi:10.1109/ICCI61671.2024.10485133

Anomaly detection history: techniques, t. a. (2021). *Chaos Genius - Blog | Explore Databricks & Snowflake Tips.* Retrieved from https://www.chaosgenius.io/blog/a-brief-history-of-anomaly-detection/

Araujo, I., Antunes, N., & Vieira, M. (2023). Evaluation of machine learning for intrusion detection in microservice applications. *Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, (pp. 126–135). Retrieved from https://dl.acm.org/doi/10.1145/3615366.3615375

Asad, H., Adhikari, S., & Gashi, I. (2024). A perspective–retrospective analysis of diversity in signature-based open-source network intrusion detection systems. *International Journal of Information Security*, 1331-1346. Retrieved from https://doi.org/10.1007/s10207-023-00794-9

Chalapathy, R., & Chawla, S. (2019). *DEEP LEARNING FOR ANOMALY DETECTION: A SURVEY.* Retrieved from https://arxiv.org/pdf/1901.03407

Díaz-Verdejo, J., Muñoz-Calle, J., Alonso, R. E., & Madinabeitia, G. (2022). On the detection capabilities of signature-based intrusion detection systems in the context of web attacks. *Applied Sciences*. doi:10.3390/app12020852

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning*, (pp. 1126-1135). Retrieved from https://proceedings.mlr.press/v70/finn17a.html

Hasan, B., & Abdulazeez, A. (2021). A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*. Retrieved from https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/8032

Kale, R., Lu, Z., Fok, K., & L. L. Thing, V. (2022). A hybrid deep learning anomaly detection framework for intrusion detection. *2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, (pp. 137-142). Retrieved from https://ieeexplore.ieee.org/abstract/document/9799486

Kosińska, J., & Tobiasz, M. (2022). Detection of cluster anomalies with ml techniques. *IEEE*. Retrieved from https://ieeexplore.ieee.org/abstract/document/9925210

Moon, J., Noh, Y., Jung, S., Lee, J., & Hwang, E. (2023). Anomaly detection using a model-agnostic meta-learning-based variational auto-encoder for facility management.

Journal of Building Engineering. Retrieved from https://www.sciencedirect.com/science/article/pii/S2352710223002784

Musa, U., Chhabra, M., Ali, A., & Kaur, M. (2020). Intrusion detection system using machine learning techniques: a review. *2020 International Conference on Smart Electronics and Communication (ICOSEC).* Retrieved from https://ieeexplore.ieee.org/abstract/document/9215333

Nassif, A., Talib, M., Nasir, Q., & Dakalbab, F. (2021). Machine learning for anomaly detection: a systematic review. *IEEE Access*. Retrieved from https://ieeexplore.ieee.org/abstract/document/9439459

On the application of principal component analysis to classification problems. (2021). *Data Science Journal*. Retrieved from https://datascience.codata.org/articles/10.5334/dsj-2021-026

PANG, G., SHEN, C., CAO, L., & HENGEL, A. (202). Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv. 54, 2, Article 38.* ACM. Retrieved from https://arxiv.org/pdf/2007.02500

Sever, Y., & Dogan, A. H. (2023). A Kubernetes dataset for misuse detection. doi:10.52953/fplr8631

Silva, M., Daniel, S., Kumarapeli, M., Mahadura, S., Rupasinghe, L., & Liyanapathirana, C. (2022). Anomaly detection in microservice systems using autoencoders. *2022 4th International Conference on Advancements in Computing (ICAC)*, (pp. 488-493). Retrieved from https://ieeexplore.ieee.org/abstract/document/10025259

Song, X., Liu, Y., Xue, L., Wang, J., Zhang, J., Wang, J., . . . Cheng, Z. (2020). Time-series well performance prediction based on Long Short-Term Memory (Lstm) neural network model. *Journal of Petroleum Science and Engineering*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0920410519311039

Tien, C.-W., Huang, T.-Y., & Kuo, S.-Y. (2020). KubAnomaly: Anomaly detection for the Docker orchestration platform with neural network approaches. *Engineering Reports*. Retrieved from https://onlinelibrary.wiley.com/doi/10.1002/eng2.1208

Tuli, S., Casale, G., & Jennings, N. (2022). *Tranad: deep transformer networks for anomaly detection in multivariate time series data.* Retrieved from http://arxiv.org/abs/2201.07284

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. *Neural Information Processing Systems*. Retrieved from https://research.google/pubs/attention-is-all-you-need/

Wang, X., Pi, D., Zhang, X., Liu, H., & Guo, C. (2022). Variational transformer-based anomaly detection approach for multivariate time series. *Measurement, 191*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0263224122000914

Xu, J., Wu, H., Wang, J., & Long, M. (2022). Anomaly transformer: time series anomaly detection with association discrepancy. Retrieved from http://arxiv.org/abs/2110.02642

Yolchuyev, A. (2023). Extreme gradient boosting based anomaly detection for kubernetes orchestration. *2023 27th International Conference on Information Technology (IT).* Retrieved from https://ieeexplore.ieee.org/abstract/document/10078576

Zakariah, M., AlQahtani, S., Alawwad, A., & Alotaibi, A. (2023). Intrusion detection system with customized machine learning techniques for nsl-kdd dataset. *Computers, Materials & Continua.* CMC. Retrieved from https://www.techscience.com/cmc/v77n3/55048