Critical Evaluation of User and Entity Behavioural Analysis Techniques

Submitted in partial fulfilment of the requirements of Edinburgh Napier University for the Degree of

BSc (Hons) Cyber Security

School of Computing

April 2023

Authorship Declaration

I, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project,

I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed:

Date: 03/04/2023

Matriculation no:

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may not make this dissertation available to others.

Abstract

Insider threats pose an increasingly prevalent threat to organisations. The escalated privilege of insiders and the dilution of organisation’s perimeters, through the increase in remote working and the expansion of cloud, makes detecting insider threats challenging. A variety of products have been developed to help address this problem, including User and Entity Behavioural Analysis (UEBA) solutions. UEBA solutions apply different anomaly detection techniques to learn the baseline profile of user and entity behaviour, to allow meaningful deviations to be identified. This project critically evaluates a selection of techniques to identify the most effective for UEBA, specifically in enterprise network traffic. Concluding that out of the techniques critically evaluated, Agglomerative Clustering, an Autoencoder, an AnoGAN and an IF and OSVM Average Ensemble, the Autoencoder proved most effective.

A literature review is performed, covering key concepts and previous work relating to UEBA. Providing the theoretical underpinning of the report, and, informing the development of the domain specific methods. Furthermore, revealing that previous literature tended to focus on the analysis of one technique, or a comparison within categories of techniques. This report attempts to build on this gap by providing a comparison between categories.

The methods, including the experiment and evaluation methods are outlined. Encompassing dataset selection and pre-processing, technique development and the evaluation process. Each of the techniques are evaluated using the metrics accuracy, precision, recall, f1-score and false positive rate. With effectiveness calculated through a defined combination of these metrics.

The results showed, out of the techniques evaluated, the Autoencoder was most effective for UEBA of enterprise network traffic. However, a large variance was also seen between metrics produced when evaluating the techniques using different datasets. Supporting findings of the literature review that highlighted the importance of datasets resembling the final deployment scenario to give a true representation of a technique’s performance when deployed. Presenting the creation of datasets that allow metrics representative of the final deployment scenario to be derived, as a potential area for future work.

Table of Contents

[**List of Tables** **1**](#_Toc98198)

[**List of Figures** **2**](#_Toc98199)

[**List of Abbreviations** **3**](#_Toc98200)

[**Glossary of Terms** **4**](#_Toc98201)

[**Acknowledgements** **6**](#_Toc98202)

[1 INTRODUCTION 7](#_Toc98203)

[1.1 Context and Background 7](#_Toc98204)

[1.2 Aim and Objectives 8](#_Toc98205)

[1.3 Research Questions 9](#_Toc98206)

[1.4 Summary 9](#_Toc98207)

[2 LITERATURE REVIEW 11](#_Toc98208)

[2.1 Introduction 11](#_Toc98209)

[2.2 Insider Threats 11](#_Toc98210)

[2.3 Concept of User and Entity Behavioural Analysis (UEBA) 11](#_Toc98211)

[2.4 Previous Work on UEBA 13](#_Toc98212)

[2.5 Review of Techniques for UEBA 13](#_Toc98213)

[2.6 Review of Recommendations for UEBA Technique Testing and Evaluation 17](#_Toc98214)

[2.7 Technologies for Implementation of UEBA Techniques 18](#_Toc98215)

[2.8 Review of Available Data Sets 19](#_Toc98216)

[2.9 Review of Evaluation Metrics 21](#_Toc98217)

[2.10 Conclusion of Literature Review 23](#_Toc98218)

[3 METHODODOLOGY 25](#_Toc98219)

[3.1 Introduction 25](#_Toc98220)

[3.2 Literature Review 25](#_Toc98221)

[3.3 Experiment Method 26](#_Toc98222)

[3.4 Evaluation Method 32](#_Toc98223)

[3.5 Project Management Method 33](#_Toc98224)

[4 RESULTS AND DISCUSSION 34](#_Toc98225)

[4.1 Introduction 34](#_Toc98226)

[4.2 Experiment Experience 34](#_Toc98227)

[4.3 Technical Artefacts 35](#_Toc98228)

[4.4 Results 35](#_Toc98229)

[4.5 Discussion 37](#_Toc98230)

[4.6 Conclusion 38](#_Toc98231)

[5 CONCLUSION 39](#_Toc98232)

[5.1 Introduction 39](#_Toc98233)

[5.2 Project Overview 39](#_Toc98234)

[5.3 Project Evaluation 40](#_Toc98235)

[5.4 Limitations 42](#_Toc98236)

[5.5 Future Work 43](#_Toc98237)

[5.6 Self-Appraisal 44](#_Toc98238)

[6 REFERENCES 45](#_Toc98239)

[7 APPENDICES 51](#_Toc98240)

[7.1 Appendix A: Data Pre-Processing 51](#_Toc98241)

[7.2 Appendix B: Feature Selection 54](#_Toc98242)

[7.3 Appendix C: Testing and Training Data 56](#_Toc98243)

[7.4 Appendix D: Model Definition 59](#_Toc98244)

[7.5 Appendix E: Evaluation Metrics 61](#_Toc98245)

[7.6 Appendix F: Individual Model Results 63](#_Toc98246)

# List of Tables

**Table 1** Review of Techniques from Literature ..........................................................................12

**Table 2** Summary of Recommendations ...................................................................................18

**Table 3** Properties of Reviewed Datasets .................................................................................20

**Table 4** Explanation of True and False Negative and Positive Rates ........................................21

**Table 5** Explanation of Evaluation Metrics ...............................................................................22

**Table 6** Justification of Technique Selection .............................................................................23

**Table 7** Summary of steps to improve reliability and validity and reduce bias and error ............28

**Table 8** Criteria for Data Formatting Steps ................................................................................30

**Table 9** Model Parameter Justification ......................................................................................31

**Table 11** Average Metrics and Effectiveness for each Model ....................................................36

**Table 12** BR of the test set of each dataset ..............................................................................37

Table 13 Average Effectiveness across Datasets .....................................................................37

**Table 15** Pre-processing steps to be applied to each dataset. ..................................................51

**Table 16** Training and Test Set Details .....................................................................................56

**Table 17** Data sets size and split details ...................................................................................57

**Table 18** Functions to be used to build each model. .................................................................59

**Table 19** Results of the Evaluation of the Autoencoder. ............................................................63

**Table 20** Results of the Evaluation of the IF and OSVM Ensemble. .........................................63

**Table 21** Results of the Evaluation of the AnoGAN. .................................................................64

**Table 22** Results of the Evaluation of the Agglomerative Clustering. ........................................64

# List of Figures

**Figure 1** Project Management Gantt Chart ...............................................................................10

**Figure 2** Visualisation from the experiment by Babu (2020) ......................................................14

**Figure 3** Results from Le and Zincir-Heywood (2020) Experiment ............................................16

**Figure 5** Research Methodology Visualisation ..........................................................................26

**Figure 6** Experiment Methodology Flowchart ...........................................................................27

**Figure 7** Evaluation Method Flowchart .....................................................................................33

**Figure 8** Experiment and Evaluation Methods Time Management ............................................33

**Figure 9** Code for CICIDS2017 Pre-processing ........................................................................52

**Figure 10** Code for UNSW-NB15 Pre-Processing ....................................................................53

**Figure 11** Code for calculating Mutual Information ...................................................................54

**Figure 12** Top 20 Important Features of CICIDS2017 ..............................................................54

**Figure 13** Top 20 Important Features of UNSW-NB15 .............................................................55

**Figure 14** Code for splitting CICIDS2017 into its respective sets. .............................................58

**Figure 15** Code for splitting UNSW-NB15 into its respective sets. ............................................58

**Figure 16** Code for the Autoencoder. .......................................................................................59

**Figure 17** Code for the AnoGAN...............................................................................................59

**Figure 18** Code for the IF and OSVM Ensemble. .....................................................................60

**Figure 19** Code for the Agglomerative Clustering. ....................................................................60

**Figure 20** Code for the evaluation metrics. ...............................................................................61

**Figure 21** Code for evaluating the Autoencoder. ......................................................................61

**Figure 22** Code for evaluating the Agglomerative Clustering. ...................................................62

**Figure 23** Code for evaluating the AnoGAN. ............................................................................62

**Figure 24** Code for evaluating to IF and OSVM Ensemble. ......................................................62

# List of Abbreviations

|  |  |
| --- | --- |
| **ADAM** | Audit Data Analysis and Mining |
| **AE** | Autoencoder |
| **AI** | Artificial Intelligence |
| **AOC** | Area under the Curve |
| **BR** | Base-rate |
| **CMSCE16** | Comprehensive, Multi-Source Cyber-Security Events 2016 Dataset |
| **FN** | False Positive |
| **FP** | False Negative |
| **FPR** | False Positive Rate |
| **GAN** | Generative Adversarial Network |
| **IDS** | Intrusion Detection System |
| **IF** | Isolation Forest |
| **LERAD** | Learning Rules for Anomaly Detection |
| **LODA** | Lightweight On-Line Anomaly Detection |
| **LOF** | Local Outlier Factor |
| **MINDS** | Minnesota Intrusion Detection System |
| **ML** | Machine Learning |
| **OSVM** | One-class Support Vector Machine |
| **ROC** | Receiving Characteristic Curve |
| **SIEM** | Security Information and Event Management |
| **TN** | True Negative |
| **TP** | True Positive |
| **UBA** | User Behavioural Analysis |
| **UEBA** | User and Entity Behavioural Analysis |
| **UNH17** | Unified Host and Network 2017 Dataset |

# Glossary of Terms

|  |  |
| --- | --- |
| Term |  |
| Agglomerative Clustering | A form of hierarchical clustering that groups data based on its similarities. |
| Artificial intelligence (AI) | A field dedicated to the development of machines which can show intelligence. |
| Autoencoder | An artificial neural network comprising of and encoder and decoder that allows it to learn the important features of data so that it can reconstruct it from a compressed version. |
| Base-rate (BR) | The proportion of positive samples in a set of data. |
| Dataset | A collection made of one or more records of related data. |
| Deep Learning | A subset of machine learning, reached when a model has multiple hidden layers. |
| Ensemble methods | Models which aim to improve performance through combining classifiers. |
| Enterprise Network | A computer network infrastructure used by medium to large organisations. |
| False Positive Rate (FPR) | The proportion of normal data that is incorrectly tagged as being not normal. |
| Generative Adversarial Network (GAN) | A deep-learning based generative model comprising of two neural networks, a generator and discriminator, that compete against each other to generate real looking data. |
| Insider threat | A threat posed by an individual who has internal access to an organisation’s assets. |
| Intrusion Detection systems (IDS) | A device or software application used to monitor systems or networks for suspected malicious activity. |
| Machine Learning (ML) | A subset of artificial intelligence that provides machines with the ability to learn without being explicitly programmed. |
| Network Traffic | The data that flows across a computer network. |
| Security Information and Event Management  (SIEM) | A technology which aggregates and analyses data from many sources to identify potential threats. |
| Supervised Techniques | Machine learning models that are taught to recognise specific, known, patterns from labelled data. |
| Unsupervised Techniques | Machine learning models used to identify unknowns and learn patterns from unlabelled data. |
| User and entity behaviour analysis (UEBA) | Analytics used to detect anomalies in the behaviour of users and entities. |

# Acknowledgements

I would like to thank and acknowledge the following people who supported me throughout this project, for their encouragement, time and feedback.

Firstly, I would like to thank John Colvin for providing continuous support, guidance and insight throughout the project.

Additionally, I would like to thank the Edinburgh Napier School of Computing for all the resources provided to help support the development of the project.

Furthermore, thank you to my family for their support and encouragement.

# Introduction

## Context and Background

Insider threats refer to threats posed, either through negligence or malicious intent (Tian, et al.,

2020), by individuals who have internal access to an organisation’s assets (Saxena, et al., 2020). Their level of privilege allows their activities to go easily unnoticed (Singh, Mehtre, & Sangeetha, 2019) and makes their detection complex (Le & Zincir-Heywood, 2021). Furthermore, increase in remote working and the expansion of cloud, has caused organisation’s perimeters to become less distinct, adding to the difficulty of detecting insider threats (Alshab, et al., 2021). Current reporting (Ponemon Institute, 2022) shows the frequency of insider-related incidents has increased by 44% over the last two years, whilst literature has consistently highlighted Insider threats as a prevalent security threat to organisations (Datta, Dasgupta, Dasgupta, & Reddy, 2021).

A variety of products have been developed to combat insider threats, with Gartner’s Market Guide (Care, Furtado, & Predovich, 2022) presenting the various solutions available. Including User and Entity Behavioural Analysis (UEBA) solutions, Security Information and Event Management (SIEM) software and wide-ranging data analytics platforms. Moreover, in their market analysis, Gartner highlight the expectation from buyers that solutions will provide UEBA functionality even if they aren’t dedicated UEBA solutions.

UEBA solutions monitor user and entity behaviour, learning normal behaviour profiles to allow deviations to be flagged as potential anomalies (Liu H. , 2021), enabling early detection of insider threats. Within UEBA solutions, lies an analytical engine wherein the standard profiles are developed and suspected anomalies are identified (Shashanka, Shen, & Wang, 2016). Achieved through running different analytics on a range of data, commonly including network traffic. Analytics can range from visualisation and statistical techniques (Babu & R, 2021) to advanced, artificial intelligence (AI) techniques (Le & Zincir-Heywood, 2021). Furthermore, deviations can be identified from a comparison of users’ behaviour to, their historical behaviour, that of their peers, or a combination of both (Shashanka, Shen, & Wang, 2016). Profiling based on historical behaviour involves creating a picture of what looks normal for a user, based on their past behaviour. In contrast, peer profiling creates the definition through learning the normal behaviour of a group of similar users or entities. The suitability of each profiling method is dependent on the scenario in which it is applied.

Despite the wide range of techniques presented in literature, a large discrepancy exists between the volume of solutions in literature and their applications in practice. Furthermore, little literature provides comparison between categories of techniques. This paper aims to address this by evaluating techniques from different categories.

## Aim and Objectives

### Aim

The aim of this project is to critically evaluate a selection of techniques for UEBA to identify the most effective technique for analysing user and entity behaviour, specifically in enterprise network traffic.

Achieving this aim will allow effective techniques for the analytical engine within a UEBA solution to be identified. Providing the foundations for a UEBA solution that can be deployed to monitor network traffic and identify anomalous behaviour.

The techniques to be evaluated will be selected based on the findings of the literature review. The criteria for selection being techniques observed to have performed most effectively for UEBA analytics or that have been earmarked as promising emerging solutions in the market. Furthermore, the measure of effectiveness will be defined through metrics, also identified in the literature review, which have proven to provide an accurate and valid measure of a technique’s ability to profile normal behaviour and allow meaningful deviations to be identified.

### Objectives

The aim will be achieved through the completion of the objectives outlined below.

1. Review contemporary literature from diverse academic fields relating to recent work on UEBA, analysis techniques, and, technologies, datasets and metrics for implementing and evaluating techniques, to inform methodology and analysis.
2. Assess the findings of the literature review to identify a selection of techniques, that have successfully been used in UEBA analytics or that have been earmarked as promising emerging solutions in the market, to be evaluated as part of the experiment.
3. Create, develop and test a methodology for developing, implementing and evaluating the selected techniques, using the knowledge gained during the literature review.
4. Execute the evaluation of each technique and gather the performance metrics.
5. Critically evaluate the results of the experiment to identify the most effective technique.
6. Draw conclusions and make recommendations for future work.

## Research Questions

* Which technique will be able to identify anomalies most effectively?
* Which metrics will be appropriate for the evaluation of the techniques?
* Which datasets will be appropriate for the evaluation of the techniques?

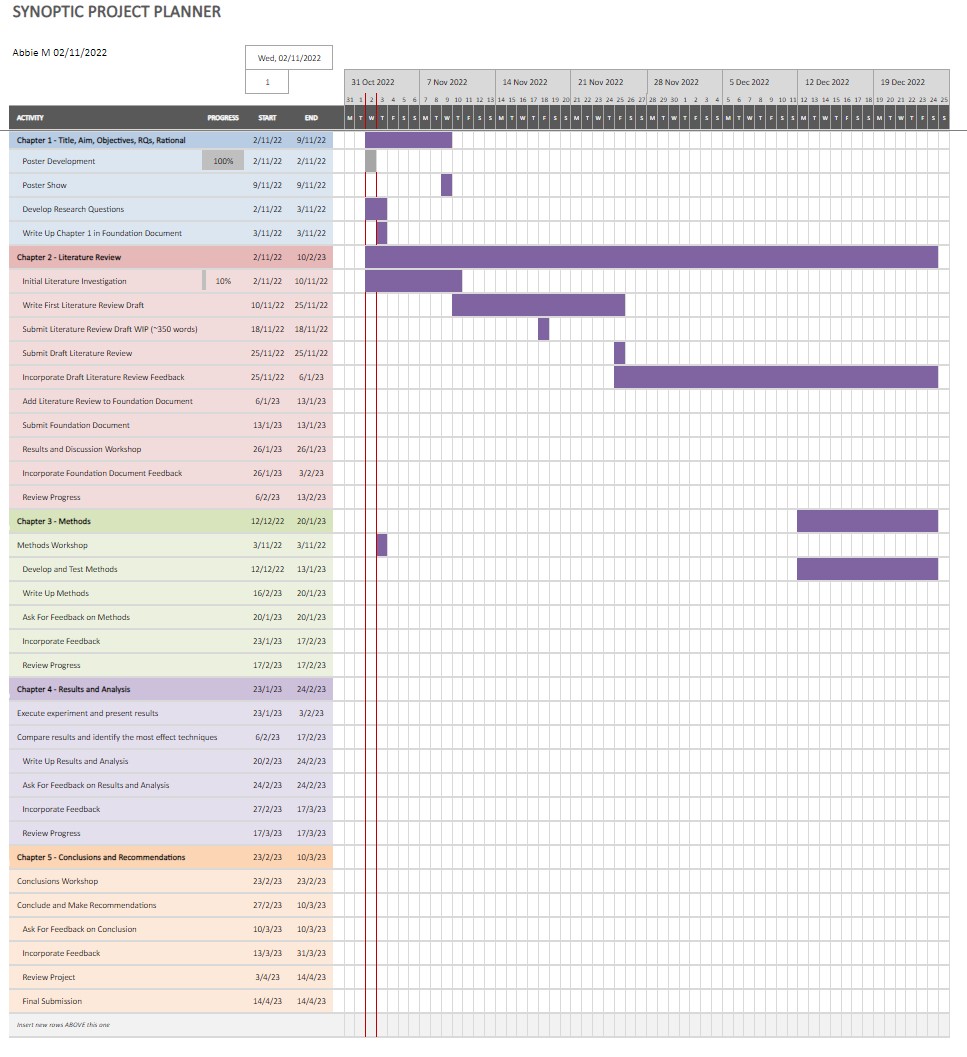
## Summary

The project will comprise of an initial literature review whereby a diverse range of sources will be critically analysed to present the relevant theoretical concepts. Based on the findings of the literature review, techniques will be selected for evaluation. Furthermore, a methodology, underpinned by the literature review, for evaluating the techniques, will be created, developed, and tested. The experiment will be executed and performance metrics gathered for each technique. Critical evaluation of the results will be performed to identify the most effective technique. Conclusions will be drawn from the results and the experience of the process, and recommendations will be made.

To support the management of the project, a Gantt chart, shown in Figure 1, will be used to schedule relevant activities so that sufficient time is allotted for each section of the report.

Furthermore, it will allow progress to be tracked, enabling better awareness of the project’s development and deadlines.

**Figure 1** *Project Management Gantt Chart*



Note. Created by author 2nd November 2022.

To keep the project effective, the scope will be limited to evaluating techniques for UEBA of enterprise network traffic, with the selection of techniques informed by the literature review. Reducing the scope to enterprise network traffic keeps the project relevant to both current research and workplace objectives, whilst directing the focus on the techniques. Open-source datasets of enterprise network traffic will be used to test and evaluate the techniques. The datasets will be chosen based on the literature review findings, the techniques chosen to be evaluated, and their academic reputation.

# Literature Review

## Introduction

This literature review aims to build a foundation of knowledge to support the project and inform the methods and analysis, through exploring the problem, theoretical concepts, and existing work in the related fields of research. A thematic approach will be adopted, reviewing contemporary literature covering the concept of insider threats and UEBA, previous research in the area, reviewing techniques for UEBA followed by recommendations for successfully implementing them, technologies for implementation, datasets, and metrics for evaluating techniques.

## Insider Threats

Saxena et al. (2020) define insider threats as those posed by individuals who have internal access to an organisation’s assets, with Tian et al. (2020) adding this can be through negligence or malicious intent. Singh, Mehtra and Sangeetha (2019) explain that insiders have an escalated level of privilege, making the detection of malicious activity challenging. One way of identifying insider threats, presented by Saxena et al. (2020), is to gather and monitor indicators of compromise. One of these indicators being anomalous behaviour, identified through anomaly detection, whereby normal patterns of behaviour are learned and deviations are flagged for investigation. For example, an anomaly detection system may be used to monitor the size of data leaving a network. If a user sends an abnormally large packet out the network this would be flagged to allow analysts to investigate whether the activity was malicious or not. Datta et al. (2021) explains that UEBA builds on this concept by bringing the focus onto both the behaviour of users and entities.

## Concept of User and Entity Behavioural Analysis (UEBA)

According to Babu (2020), UEBA is a security paradigm whereby baseline profiles of user and entity behaviour are developed and tracked, allowing deviations to be highlighted as potential anomalies. Providing an effective solution for identifying insider threats, as demonstrated by Khaliq et al. (2020), which Shashanka et al. (2016) highlight as a key problem in enterprise networks. Furthermore, enabling early detection of threats, which Rashi and Miri (2021) argue minimises their impact.

UEBA solutions harness various analytical techniques, demonstrated in the variety of techniques observed across literature, shown in Table 1. Ranging from basic visualisation techniques as demonstrated by Babu (2020) to complex ensemble methods such as those presented by Le and Zincir-Heywood (2021). Liu (2021) argues these techniques enable UEBA solutions to overcome the static nature of traditional, rule-based solutions. Allowing the detection of unknowns that escape traditional approaches. Furthermore, Datta et al. (2021) explain UEBA solutions widen the scope of earlier User Behavioural Analysis (UBA) approaches by encompassing entities, such as endpoints, data repositories, and routers, to give a better-rounded analysis.

**Table 1** *Review of Techniques from Literature*

|  |  |  |
| --- | --- | --- |
| Category of Techniques Applied / Researched | | Sources |
| AI-Based | Machine Learning | (Datta, Dasgupta, Dasgupta, & Reddy, 2021)  (Le & Zincir-Heywood, 2021)  (Lukashin, Popov, Bolshakov, & Nikolashin,  2019)  (Mehta, Kothuri, & Garcia, 2018)  (Savenkov & Ivutin, 2020)  (Slipenchuk & Epishkina, 2019) |
| Deep Learning | (Jallad, Aljnidi, & Desouki, 2020)  (Le & Zincir-Heywood, 2021)  (Liu H. , 2021)  (Sabuhi, Zhou, Bezemer, & Musilek, 2021)  (Savenkov & Ivutin, 2020)  (Sun, Guo, Li, Xu, & Wang, 2019)  (Tian, et al., 2020) |
| Ensemble | (Diop, Emad, Winter, & Hilia, 2019)  (Le & Zincir-Heywood, 2021)  (Liu H. , 2021)  (Singh, Mehtre, & Sangeetha, 2019)  (Singh & Srivastav, 2021) |
| Non-AI-Based | Visualisation | (Babu & R, 2021)  (Babu S. , 2020) |
| Statistical | (Legg, Buckley, Goldsmith, & Creese, 2017)  (Mehta, Kothuri, & Garcia, 2018)  (Shashanka, Shen, & Wang, 2016) |

Note. Created by author on 7th December 2022.

## Previous Work on UEBA

The concept of UEBA was derived from Denning’s (1987) landmark paper which first introduced anomaly detection for intrusion detection. Following Denning’s paper, a variety of different models for intrusion detection systems (IDS) were developed, many of which were featured in Lazarevic’s (2005) survey. Including the Minnesota IDS (MINDS) developed by Ertoz et al. (2004), the Audit Data Analysis and Mining (ADAM) IDS demonstrated by Barbara et al. (2001) and the Learning Rules for Anomaly Detection (LERAD) algorithm created by Mahoney and Chan (2003). However, multiple of these solutions, including those by Ertoz et al. (2004), Barbara et al. (2001) and Mahoney and Chan (2003), were presented with scepticism from critics such as Gates and Taylor (2006).

Gates and Taylor (2006) felt the concepts introduced by Denning were being applied incorrectly, without regard for the context in which they were being used and without consideration of the original issues highlighted by Denning. One of their arguments being that Denning made the assumption that attacks are rare and anomalous, which may have stood true at the time of writing, however with the evolution of computing and networking environments, attackers have found ways to hide their attacks and attacks are no longer rare. Gates and Taylor (2006) argued that systems such as MINDS by Ertoz et al. (2004), which were based on the outdated assumption that all attacks are anomalous and a small proportion of traffic, should take more consideration of the ways in which attackers can hide their behaviour and the current proportion of attacks seen in traffic. Following this, more papers could be seen acknowledging the issues highlighted by Gates and Taylor, one of these being the paper written by Shashanka et al. (2016), one of the first papers to mention UEBA, who ensured their assumptions aligned with the context of their solution.

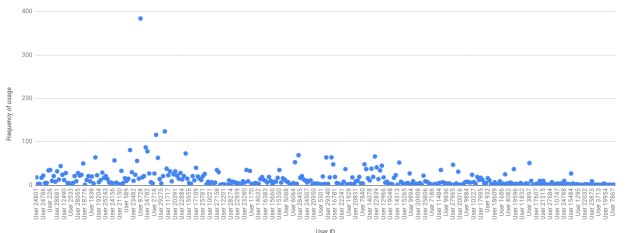
Since then, research into UEBA solutions has grown, and, as highlighted by Khaliq et al. (2020) many commercial UEBA solutions have emerged on the market, however at their time of writing Open-source solutions still lagged behind. Recently, however, a new open-source solution, OpenUEBA, was presented by Calvo (2022), potentially marking advancements in the area since 2020. Moreover, as shown in Table 1 a wide range of techniques for UEBA analytics have been presented in literature in recent years.

## Review of Techniques for UEBA

When comparing the UEBA solutions presented in literature, such as the older surveys, for example, those by Liu et al. (2018) and Fernandes et al. (2018) to newer surveys such as those by Kaliq et al. (2020) and Martin et al. (2021) an emergent use of AI-based techniques can be seen.

The strength of AI techniques in comparison to more basic techniques, such as visualisation, can be observed when reviewing papers such as that by Babu (2020). Whereby anomalies in network traffic were detected through visualisations and simple statistical modelling. For Example, in Figure 2, Babu grouped users by their usage to identify outliers such as User8728. Although this provided a quick insight into access frequency, as previously discussed when reviewing the paper of Gates and Taylor (2006), it cannot be assumed that an anomaly indicates exploitative behaviour. Furthermore, without knowing the user’s normal behaviour, the context of their activity cannot be understood. Contrastingly, as explained by Khaliq et al. (2020), AI techniques address this by learning the baseline behaviours so that only deviations to the normal are flagged, with the normal being unique to a user, entity or group.

**Figure 2** *Visualisation from the experiment by Babu (2020)*



Note. Reprinted from Babu, S. (2020). Detecting Anomalies in Users - An UEBA Approach. In Proceedings of the International conference on Industrial engineering and Operations Management, Dubai, UAE, March 10-12, 2020, (pp.

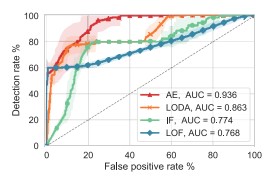
863–876).

Ongsulee (2019) gives a brief explanation of AI, describing it as the intelligence shown by a machine, a subset of which is machine learning (ML). In ML, a machine learns without being told explicitly what to know. ML is generally divided into two methods: supervised and unsupervised. In terms of UEBA, Kaliq et al. (2020) explain that supervised techniques use labelled datasets, where data is tagged with labels the model should learn, for example, anomalous or notanomalous, to teach a model to recognise anomalies. Contrastingly, unsupervised techniques use unlabelled datasets to detect unknowns. Ongsulee (2019) explains further that ML also has a subset called deep learning, reached when a model has multiple hidden layers.

Unsupervised learning techniques appear more frequently in UEBA analytics presented in literature. Datta et al. (2021) compared the performance of different unsupervised learning algorithms for UEBA and attempted to identify, from the results, which algorithms would be best combined for insider threat detection. K-means and Agglomerative Clustering were presented as the best combination for insider threat detection. However, the resilience of these metrics when applied to a real-world scenario could be questioned as the experiment was conducted on a dataset of click-stream data for an online shop. This does not resemble data the methods would be applied to in the real-world. From papers such as that by Heine, Laue and Kleiner (2020), the importance of the datasets used to evaluate techniques is highlighted. With datasets not resembling the final deployment scenario often being featured in the reasons for the discrepancy between evaluation results of UEBA solutions presented in papers compared with those observed in practice.

Le and Zincir-Heywood (2021) also compared unsupervised algorithms for insider threat detection, however, in contrast to Datta et al. (2021), they evaluated the algorithms on a selection of datasets of network traffic. Making their investigation more relevant to insider threat detection. As part of their investigation, they compared an Autoencoder (AE), Isolation Forest (IF), Local Outlier Factor (LOF) and Lightweight On-Line Anomaly Detection (LODA). Their results, shown in Figure 3, showing the AE performed best at very low False Positive Rates (FPR), the quantity of normal data classified as anomalous. Jallid et al. (2020) highlight FPR as a crucial factor in anomaly detection as a high FPR is often the reason for techniques not being deployed in practice.

**Figure 3** *Results from Le and Zincir-Heywood (2020) Experiment*



Note. Reprinted from Le, D. C., & Zincir-Heywood, N. (2021). Anomaly Detection for Insider Threats Using Unsupervised Ensembles. IEEE Transactions on Network and Service Management, 1152-1164. doi:10.1109/TNSM.2021.3071928.

Furthermore, Le and Zincir-Heywood (2021) built on their review by testing the combination of the algorithms in different ensemble methods. Ensemble methods, as explained by Diope et al. (2019), combine weak classifiers for improved results. Their investigation into VOTE2,3,4, MAX, and AVG ensembles revealed VOTE2,3 and AVG significantly outperformed the VOTE4 ensemble and the individual classifiers. Except the AE, whose performance was on par with the VOTE2,3 and AVG ensembles. Sing and Srivstav (2021) also utilised an ensemble in their proposed deep learning ensemble framework. Involving Isolation Forest (IF) and One-class Support Vector Machine (OSVM) algorithms. They tested the framework on a network traffic dataset, also used by Le and Zincir-Heywood. Whilst evaluating the framework they were able to achieve a high performance compared to other frameworks in literature at their time of writing. Further validating the effective nature of ensembles for UEBA analytics.

Alongside ensemble methods, Generative Adversarial Networks (GAN) have also proved to be effective deep-learning techniques for UEBA. Sabuhi et al. (2021) conducted a systematic literature review on the use of GANs for anomaly detection. Explaining that GANs are generative, not discriminative, meaning that rather than identifying the most suitable label for an instance, they identify the likelihood of an instance based on previous instances. In the review, Sabuhi et al. (2021) highlight the success of GANs for anomaly detection in network traffic and present the types of GANs that have been observed being used in literature. Explaining that most anomaly detection techniques using GANs for representation learning are varieties of the AnoGAN technique, the first GAN-based anomaly detection technique. AnoGAN is built on the DCGAN architecture, however an additional step is added after training the generator to allow an anomaly score to be computed between a piece of data and the most similar data generated by the generator. With anomalies then defined based on a threshold anomaly score.

Reviewing the techniques presented in literature has shown an emergent adoption of AI-based techniques, with a lean towards unsupervised techniques. High performance has been observed being achieved with deep-learning models, in particular, Autoencoders, Ensembles and GANs. Furthermore, it can be observed that literature has tended to focus on comparing techniques within categories, for example a comparison of different GANs. However little literature provides comparison between categories such as ensembles and GANs.

## Review of Recommendations for UEBA Technique Testing and Evaluation

Gates and Taylor (2006) produced one of the first papers highlighting the discrepancy between the large volumes of research going into anomaly detection systems compared to their uptake in practice. With the gap still being acknowledged in more recent papers such as that by Heine, Laue and Kleiner (2020), wherein they attempted to identify the reasons for the discrepancy and present recommendations for improving it.

The first set of recommendations made by Heine, Laue and Kleiner (2020) highlights the importance of using suitable datasets. They state the dataset should contain traffic from a network that closely resembles the final deployment scenario, else the metrics may be invalid. Furthermore, data should be diverse enough to prevent models from simply using features such as IP and port combinations to identify anomalies. Moreover, features in the dataset should also be present in the network in which the solution will be deployed, else the model will not be transferrable. Additionally, highlighting that when splitting datasets for testing and training they should be split to include a consecutive time span or else the typical pattern of a period would be corrupted, which is further supported by Ring et al. (2019). Fernandes et al. (2018) further highlight the issues surrounding unsuitable datasets, stating that it was the most reoccurring issue they came across in their survey of papers on anomaly detection.

As well as setting out the recommendations regarding datasets, Heine, Laue and Kleiner (2020) also focus on the importance of choosing suitable evaluation metrics. Stating that metrics such as accuracy, frequently observed being used in literature, can show high performance whilst hiding poor factors such as high FPR. They recommend choosing metrics that are specific to the techniques being used and which show a well-rounded view of performance.

A summary of the recommendations identified is included in Table 2 below.

**Table 2** *Summary of Recommendations*

|  |  |
| --- | --- |
| **Recommendation** | **Source(s)** |
| **Dataset traffic should resemble that of the final deployment setting.** | (Heine, Laue, & Kleiner, 2020)  (Fernandes, Radrigues, Carvalho, Al-Muhtadi, & Proenca, 2018)  (Ring, Wunderlich, Sheuring, Landes, & Hotho,  2019) |
| **Datasets must be diverse enough to prevent reliance on IP and Port values.** | (Heine, Laue, & Kleiner, 2020) |
| **Features used for training must be present in the final deployment setting.** | (Heine, Laue, & Kleiner, 2020) |
| **The datasets must be split appropriately to include consecutive time spans.** | (Heine, Laue, & Kleiner, 2020)  (Ring, Wunderlich, Sheuring, Landes, & Hotho,  2019) |
| **Evaluation metrics must be specific to techniques.** | (Heine, Laue, & Kleiner, 2020) |

Note. A table summarising the recommendations identified in the review of literature. Created by author 7th December 2022.

## Technologies for Implementation of UEBA Techniques

Python offers a multitude of libraries for implementing UEBA techniques. Many of which can be observed being utilised in experiments across literature. Such as in the paper by Le and ZincirHeywood (2021) in which they used the three most common libraries observed, Scikit-Learn, PyOD and TensorFlow.

PyOD, as explained by Rashid and Miri (2021), who utilised it for UEBA analytics, is a python library dedicated to outlier detection in multivariate data. Out of the three libraries, PyOD is the newest, first introduced in the paper by Zhao et al. (2019). However, as highlighted by Rashid and Miri (2021), it has managed to gain credibility and has been successfully used to implement solutions in commercial and research projects.

Contrastingly Scikit-Learn is a more established Python Library. Introduced in the paper by Pedregosa et al. (2011). It has a broader scope than PyOD, focusing on making ML techniques accessible to a wider range of users. Providing a huge selection of contemporary ML algorithms for medium-scale problems. Datta et al. (2021) used the library to implement the algorithms they compared. Presenting the code used in their paper, revealing the simple nature of the implementation.

Similar to Scikit-learn, TensorFlow is also a well-established Python Library. However as explained by Abadi et al. (2016), it focuses predominately on deep neural networks. Examples of which can be observed in literature such as in the paper by Sun et al. (2019) in which TensorFlow was used to implement different deep learning algorithms for UEBA analytics and the paper by Tian et al. (2020) in which they used the library to implement GANs for network anomaly detection.

## Review of Available Data Sets

As previously highlighted by Heine, Laue and Kleiner (2020), choosing a suitable dataset is crucial to the representative evaluation of UEBA techniques. Despite the variety of datasets available, exemplified by the survey of datasets by Ring et al. (2019), the aforementioned recommendations limit the range of suitable choices. As this project will focus on UEBA analytics applied to enterprise network traffic, this section of the literature review will be limited to reviewing datasets of enterprise network traffic.

Through the review of literature, four suitable datasets were identified. Including the Comprehensive, Multi-Source Cyber-Security Events 2016 (CMSCSE16) dataset introduced by Kent (2016), Unified Host and Network (UHN17) dataset proposed by Beazley et al. (2019), CICIDS2017 dataset introduced by Sharfaldin et al. (2018) and UNSW-NB15 dataset recommended by Moustafa and Slay (2015). Each dataset has different attributes which could affect the evaluation metrics produced by techniques.

Both the CICIDS2017 and UNSW-NB15 datasets are synthetic, meaning unlike CMSCSE16 and UHN17, they contain emulated, not real, network traffic. Real traffic most closely resembles a real-world deployment scenario. However, due to privacy concerns, real datasets are often heavily anonymised which Ring et al. (2019) believe is a disadvantage. Yet, a question could be asked regarding the effect anonymising network traffic has. For example, Beazley et al. (2019) explain that UHN17 is anonymised by replacing IP addresses with anonymised identifiers. This does not change the feature’s relationship to its data. Both an IP address and an anonymised identifier are unique and therefore preserve the entity’s relationship with its data. Nonetheless, Ring et al. (2019) present an additional advantage of emulated datasets in the fact that they are generally labelled which is necessary for supervised machine learning. Khaliq et al. (2020) highlight this further by showing the greater number of methods that can be utilised with labelled datasets over unlabelled.

Furthermore, the datasets also differ in terms of the format of the traffic within them. Both CMSCSE16 and UHN17 contain flow-based traffic whereas CICIDS2017 and UNSW-NB15 are packet-based. Andreas et al. (2020) explains, packet-based traffic contains the complete payload whereas flow-based traffic aggregates related packets and provides the related metadata. Furthermore, Ring et al. (2019) point out that packet-based data can be converted to flow-based however flow-based cannot be converted to packet-based. Umer, Sher and Bi (2017) argue that although packet-based data has traditionally been used more, flow-based data should be given more consideration as it is less computationally and time costly to analyse, and is also not hindered by encryption in the way that packet-based data is. Furthermore, Andres et al. (2020) demonstrate that flow and packet-based data can both be used successfully to train highperforming models, however, they each perform differently with different algorithms.

Moreover, Ring et al. (2019) explain flows can be bi- or uni-directional. Bi-directional flows encapsulate requests and replies in one flow, whereas uni-directional presents them in separate flows. Out of the datasets reviewed, only CMSCSE16 is uni-directional, the others are bidirectional. In literature, most evaluations of techniques observed, used bi-directional flows, however no clear evidence of their benefit could be found.

Despite their differences, all the datasets have been used successfully for the evaluation of UEBA analytics in literature. CICIDS2017 and UNSW-NB15 have been observed being used more frequently in literature to train and evaluate algorithms. It is presumed this may be due to the fact they are labelled which means they can support the training of both unsupervised and supervised algorithms. Table 3 presents the findings of the comparison of the datasets as well as providing sources for each in which they were observed being successfully used for UEBA analytics.

**Table 3** *Properties of Reviewed Datasets*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | CMSCSE  (Kent, 2016) | UHN  (Beazley, et al., 2019) | CICIDS2017  (Sharfaldin, Lashkari, & Ghorbani, 2018) | UNSW-NB15  (Moustafa & Slay, 2015) |
| Format | Flow | Flow | Packet | Packet |
| Direction | Unidirectional | Bidirectional | Birdirectional | Bidirectional |
| Real or Emulated | Real | Real | Emulated | emulated |
| Labelled or Unlabelled | Unlabelled | Unlabelled | Labellled | Labelled |
| Anonymised | Yes | Yes | No | No |
| Size | 130M flows | 150GM flows | 3.1M flows | 2M points |
| Collect Period | 58 days | 90 days | 5 days | 31 hours |
| Year | 2016 | 2017 | 2017 | 2015 |
| Source Demonstrating Use | (Le & Zincir-Heywood, 2021) | (Nortier, et al., 2018) | (Bakhareva, et al., 2019) | (Roy & Cheung, 2018) |

Note. Created by author 7th December 2022.

## Review of Evaluation Metrics

As previously highlighted, the metrics used for evaluation are crucial for a fair comparison and accurate representation of a technique’s potential performance in a real-world deployment. Across literature, a wide range of evaluation techniques are demonstrated, with most being based on true and false, positives and negatives.

**Table 4** *Explanation of True and False Negative and Positive Rates*

|  |  |  |
| --- | --- | --- |
| **Metric** | **Meaning** | **Example** |
| **True**  **Positive**  **(TP)** | The sum of correctly classified not normal samples. | The number of samples which were supposed to be categorised as anomalous which were. |
| **True**  **Negative**  **(TN)** | The sum of correctly classified normal samples. | The number of samples which were supposed to be categorised as normal and were. |
| **False**  **Positive**  **(FP)** | The sum of samples incorrectly classified as not normal. | The number of samples which were supposed to be categorised as normal but were categorised as anomalous. |
| **False**  **Negative**  **(FN)** | The sum of samples incorrectly classified as normal. | The number of samples which were supposed to be categorised as anomalous but were categorised as normal. |

Note. Created by author on 7th December 2022 using information from the paper by Elmasry, Akbulut and Zaim (2019).

As explained in Table 4, true and false, negatives and positives classify the predictions made by a model based on whether it successfully categorises samples. From these values, a variety of other metrics can be derived, including, FPR, precision, accuracy, recall and f1-score. Demonstrated by Heine, Laue and Kleiner (2020), and explained by Elmasry, Akbulut and Zaim (2019), Table 5 describes each of these metrics and gives their formula.

**Table 5** *Explanation of Evaluation Metrics*

**Metric**

**Description**

**Formula**

**False Positive Rate** The rate of normal samples

incorrectly classified as not 𝐹𝑃

normal. 𝐹𝑃 +𝑇𝑁

**Precision** The proportion of positive

predictions that were correct. 𝑇𝑃

𝑇𝑃 +𝐹𝑃

**Accuracy** The proportion of correctly

classified samples. 𝑇𝑃 +𝑇𝑁

𝐹𝑃 +𝑇𝑃 +𝐹𝑁 +𝑇𝑁

**Recall** The proportion of not normal

samples that were correctly 𝑇𝑃

classified. 𝑇𝑃 +𝐹𝑁

**F1-Score** The harmonic mean of

precision (P) and recall (R). 2∗ 𝑃 ∗ 𝑅

𝑃 +𝑅

Note. Created by author 7th December 2022 using information from the papers by Heine, Laue and Kleiner (2020), and Elmasry, Akbulut and Zaim (2019).

Papers such as that by Sun et al. (2019), make use of the evaluation metrics described above to evaluate models. However, in other papers, such as those by Singh and Srivastav (2021) and Le and Zincir-Heywood (2021), the metrics are combined to derive more complex metrics such as Receiving Characteristic Curve (ROC) and Area under the Curve (AOC). Le and Zincir-Heywood explain that ROC demonstrates the relationship between Recall and FPR at different decision thresholds by plotting them against each other. With AOC summarising the curve in a single numeric value by calculating the area under it.

Despite the frequent use of these metrics, Heine, Laue and Kleiner (2020) demonstrate that if focused on singularly, many of the metrics can be misleading. Highlighting that if datasets are imbalanced, accuracy can still appear high despite recall being low. Furthermore, they introduce the concept of base-rate (BR), a property of a test set signifying the proportion of positives in it, which influences the relationship between precision and FPR. Creating a fallacy first described by Axelsson (2000) as the Base-Rate Fallacy, which leads to low precision despite low FPRs if the test set’s BR is low. Heine, Laue and Kleiner (2020) therefore believe that if precision is used to evaluate a model, the results should be complemented with the BR of the test set.

## Conclusion of Literature Review

UEBA solutions derive baseline profiles of user and entity behaviour from which they flag deviations as potential anomalies. Through the review of literature, a variety of different techniques for the analytical engine within UEBA solutions were identified, with a general movement towards AI-based techniques being observed. In particular, Autoencoders, Ensembles and GANs have performed well in previous evaluations of UEBA techniques. Leading to them being selected as the techniques which will be evaluated in the experimental phase of the report, alongside Agglomerative Clustering. The chosen techniques are all unsupervised techniques, however they are still unique and each presented a strong reason for their selection, as explained in Table 6.

**Table 6** *Justification of Technique Selection*

|  |  |
| --- | --- |
| Algorithm | Reason for Selection |
| Autoencoder | The Autoencoder algorithm could be observed being consistently, successfully used for UEBA in a variety of literature. Including in the paper by Le and ZincirHeywood (2021) where it even performed on par with more complex Ensemble methods. |
| IF-OSVM  Average  Ensemble | Market Guides such as that by Gartner (2018) deemed Ensemble methods to be a potential future algorithm in UEBA analytical engines. Furthermore, in the experiment by Zincir-Heywood (2021) the average Ensemble proved to be one of the higher performing Ensembles. Additionally, for Diope et al. (2019) the IF and OSVM combination proved particularly successful. |
| AnoGAN | Alongside Ensemble methods, GANs were also earmarked as potential future algorithms in UEBA analytical engines by the Gartner (2018) market guide. Additionally, Sabuhi et al. (2019) explained that most anomaly detection techniques based on representation learning through a GAN are variation of AnoGAN. |
| Agglomerative  Clustering | Datta et al. (2021) presented successful results when combing K-Means and  Agglomerative Clustering. In order, to provide a comparison between distinct |
|  | categories Agglomerative clustering will be used to represent potential performance of clustering techniques. |

Note. Created by author 7th December 2022.

Moreover, using unsupervised techniques means they will be transferrable to the final deployment scenario as the data will not be labelled. Additionally, the review revealed that previous work tended to focus on the analysis of one technique or a comparison within a category. Little literature compared different categories of techniques such as a comparison of singular classifiers, Ensembles, clustering and GANs. Therefore, the selection of techniques chosen will attempt to build on this gap by providing a comparison between techniques of different categories.

Additionally, points presented by Heine, Laue and Kleiner (2020), and Elmasry, Akbulut and Zaim (2019) highlighted the importance of choosing evaluation metrics that are transparent, valid and accurate. Therefore, when evaluating the models, a confusion matrix showing the TP, FP, TN and FN values will first be produced. From these the metrics FPR, precision, accuracy, recall and f1score will be calculated. Adopting a similar approach to that taken by Le and Zincir-Heywood (2021). Moreover, the BR of the test sets will be displayed alongside the results for completeness. Using well-known metrics should allow the results to be easily compared to results in other literature.

As well as informing the selection of the evaluation metrics, the review also highlighted considerations to take on board during the evaluation to ensure that the metrics produced give a transparent insight into real-world performance. A selection of literature highlighted that these recommendations have often not been followed, causing a discrepancy between the volume of research on analysis techniques and their uptake in practice. Therefore, it is important these considerations are incorporated in the methodology to avoid this occurring.

Furthermore, a variety of topics to support the implementation of the techniques and their evaluation in this project, were explored in the review. Including, identifying different python libraries for implementing techniques, suitable datasets and appropriate metrics for evaluation.

This will provide the theoretical underpinning for the methods and analysis.

# Methododology

## Introduction

This section aims to explain and justify the methods followed during the development of the report. Including an explanation of the research method adopted in the literature review alongside the creation, development and implementation of the experiment and evaluation methods. Furthermore, providing an explanation of the project management approach adopted.

Creating the experimental method will involve applying the knowledge gained in the literature review to inform and justify a method for developing and implementing the selected techniques. The decisions taken during the implementation will be explained during this section. Including the selection of datasets, and technologies, to be used in the experiment, based on rationale supported by the literature review. With additional consideration taken to ensure the validity and reliability of results and the minimisation of bias and error.

Developing the evaluation method will include the creation and justification of the approach to be used to evaluate the techniques to fulfil the aim of the project of critically evaluating techniques for UEBA.

## Literature Review

As previously mentioned, the literature review took the form of a thematic review. A chronological review could have been adopted however as the aim was to explore contemporary techniques and technologies, a thematic review was better suited, as it allowed the key concepts to be explored in greater depth and past work to be reviewed.

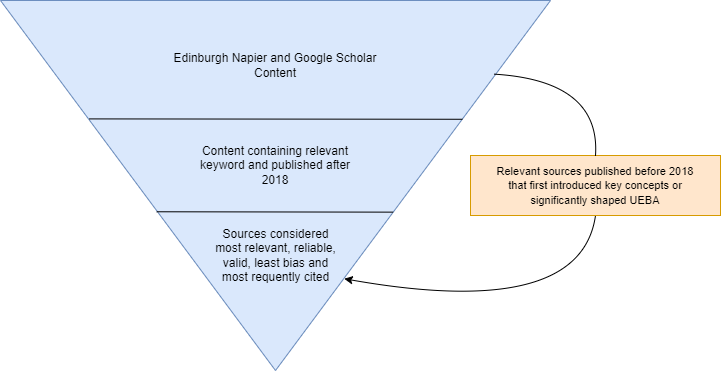
The method followed for the literature review, visualised in Figure 5, was as follows:

1. Search platforms such as Google Scholar and Edinburgh Napier University library for keywords.
2. Filter results by year of publication.
3. Identify the most relevant sources by evaluating the relevance, reliability, validity, and bias of the content through checking:

o The publisher, to ensure they are reputable. o The citation count, to understand if the paper has been impactful. o The evaluation metrics provided, to ensure they were transparent and fair. o For commercial influence or the effect of the dataset used, to identify influence of bias.

Initially, the search filtered for papers published after 2018 to ensure they were contemporary. However, as the review developed a handful of older papers were included. These papers introduced key concepts, datasets or technologies, or presented criticism which shaped the field of UEBA significantly and therefore remained relevant.

**Figure 4** *Research Methodology Visualisation*



Note. Created by author 14th December 2022.

## Experiment Method

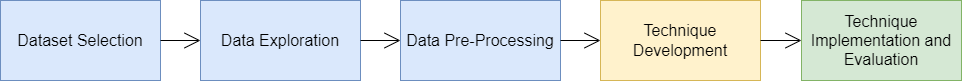
### Introduction

The aim of the experiment is to critically evaluate the selected UEBA techniques. To achieve this, the experiment will follow a process, similar to that of Le and Zincir-Heywood (2021), visualised in figure 6. The process will be as follows:

1. Select a suitable dataset based on the findings of the literature review.
2. Explore the dataset to identify the required pre-processing steps.
3. Apply the relevant pre-processing steps to the data.
4. Develop the techniques.
5. Evaluate the techniques.

Furthermore, quality issues such as reliability, validity, bias, and error will be considered prior to the development of the experimental method so that relevant steps can be incorporated.

**Figure 5** *Experiment Methodology Flowchart*



Note. Created by author 14th December 2022.

### Reliability, Validity, Bias, and Error

It is critical to ensure that the results of the experiment are reliable and valid, and that the influence of bias is minimised. Additionally, sources of error must be eliminated to ensure the experiment is successful and that the results are not influenced by error.

To ensure reliable results, steps will be taken so that the experiment can be repeated by others to produce the same results. Incorporating, including the code used to implement the techniques, specifying the development environment, documenting the data pre-processing steps and repeating the evaluation of each technique to find an average for each metric.

Furthermore, validity will be preserved by ensuring the metrics used to evaluate the techniques accurately reflect the aim of the experiment. Including, applying knowledge gained in the literature review, such as displaying the BR of the dataset alongside the results, as recommended by Heine, Laue and Kleiner (2020), to ensure the nature of the dataset is reflected. Using multiple metrics to ensure that poor performance is not hidden behind singular metrics as both Heine, Laue and Kleiner (2020) and Axelsson (2000) demonstrated was possible. Moreover using metrics that measure a techniques ability to learn normal behaviour, and from that identify potential anomalies, as this is what is being considered the effectiveness of a technique. Therefore, this will ensure the metrics are relevant to the aim of the experiment.

Additionally, the potential influence of bias has been considered. Mehrabi et al. (2021) explain bias can often arise in ML. Introduced through interaction, the data used to train a model and the algorithm within a model. In this experiment, the latter two risks present the most likely sources of bias. Mehrabi et al. (2021) state, if the dataset is non-representative of the population, representation bias can be introduced. Therefore, a dataset that contains traffic representative of an enterprise networks traffic and its general distributions, will be used. Additionally, a clear message uncovered during the literature review is that no dataset is perfect, leading Heine, Laue and Kleiner (2020) to recommend using multiple datasets. Thus, multiple datasets will be used. It is hoped this will also make the data more representative.

Finally, in order to eliminate sources of error, the datasets will be explored to identify features such as corrupt values or categorical data that may cause the model to error when presented with the data. These will then be removed or re-formatted in the pre-processing stage to ensure the data is suitable for the model.

Table 7 summarises the steps outlined above to improve reliability and validity, and reduce bias and error.

**Table 7** *Summary of steps to improve reliability and validity and reduce bias and error*

|  |  |
| --- | --- |
| Concern | Considerations |
| Reliability | Inclusion of code of each technique |
| Specifying the environment used for testing |
| Outlining data pre-processing steps |
| Repeating the evaluation to find averages |
| Validity | Displaying BR alongside results |
| Including a variety of metrics for a transparent review of performance |
| Selecting metrics relevant to the aim |
| Bias | Selecting datasets representative of an enterprise network |
| Using multiple datasets |
| Error | Identify and remove corrupt data |
| Identify and reformat incompatible values |

Note. Created by author 8th February 2023.

### Experiment Execution

#### Data Selection

During the literature review a range of recommendations were set out regarding selecting a dataset for the experiment. Furthermore, to reduce bias, it was also decided that two datasets would be used during the experiment.

The datasets selected are the UNSW-NB15 and CICIDS2017 datasets. They have been chosen as they are labelled, meaning the BR can be calculated, and the models can be evaluated, as the number of true samples is known. Using a synthetic dataset is not desirable, as they do not represent real-world traffic as accurately as a real dataset. However, this compromise will be made to uphold reliability and validity of results, by ensuring the models can be evaluated. Furthermore, the datasets have been used successfully in the past in comparable experiments such as that by Le and Zincir-Heywood (2021) and appear to contain representative data.

#### Experiment Environment

To ensure reliability of results, a consistent environment will be used for the development, training and evaluation of each model. Comprising of a notebook ran on Kaggle. Using a Kaggle notebook will also help with ease of use by providing access to the chosen datasets and required Python libraries. Furthermore, this option provides the required resources at a cost suitable for the project.

#### Data Exploration

Before the datasets can be pre-processed, they must be understood in greater detail. Including, identifying the number of files the data is split across, the features of the datasets and any values which would be incompatible with the models. Understanding this will allow the activities needed to be undertaken in the pre-processing stage to be identified.

First the datasets will need to be imported into the notebook and the files containing the network traffic must be identified. Thereafter, the data must be aggregated into one DataFrame to improve efficiency and make the data suitable for the models, as they require it in one chunk.

Then the data can be analysed to identify its features, to determine which must be removed or manipulated to make the data suitable for the model. Furthermore, to ensure the same profiling technique is used for each dataset, it must be ascertained whether unique identifiers are included in both datasets. Additionally, the labelling of the datasets must be checked to identify unsuitable columns to be removed or re-formatted when the set is split into test and training sets.

Moreover, data that is corrupt or incompatible must be identified so it can be removed as not to cause an error when the model is presented with the data. This will be achieved by filtering the DataFrame for known incompatible values. Additionally, categorical data will be identified so that it can be one-hot encoded. Furthermore, it must be determined whether the data needs to be normalised by analysing the ranges of the values to determine if they could cause bias weighting within the model.

#### Data Pre-processing

Pre-processing is required to ensure the datasets are in the specified format required for the models. This will prevent errors, and, ensure the data does not cause bias within the models and is representative of the final deployment scenario. First, the pre-processing steps identified in the data exploration stage will be applied, thereafter feature selection will be performed and the data will be split into test and training sets.

3.3.3.4.1 Dataset Formatting

As previously explained, the pre-processing steps identified during the exploration phase must first be applied to the datasets. First, if they are not present in both data sets, unique identifiers will be removed, by dropping the columns from the Dataframe. Moreover, unsuitable labelling columns will be removed or reformatted so that the label column contains only 0 for normal and

1 for anomalous traffic. Achieved by dropping the unsuitable columns and applying Panda’s functions to reformat data where necessary. Thereafter, any columns of categorical data will be one hot encoded by applying the sklearn OneHotEncoder function to them. Additionally, if it was identified that the datasets need to be normalised, the sklearn MinMaxScaler function will be applied. Table 8 provides a summary of the steps to be taken if the relevant criteria is identified in the exploration phase. Furthermore a more detailed explanation of the pre-processing steps that need to be taken for each dataset can be found in Table 15 in Appendix A.

**Table 8** *Criteria for Data Formatting Steps*

|  |  |
| --- | --- |
| Criteria | Step |
| Identifiers present in only one of the datasets. | Remove the identifiers form the dataset. |
| Unsuitable labelling present in dataset. | Reformat the label column so that it contains only 0 for normal and 1 for anomalous traffic. |
| Categorical data in dataset. | Apply the sklearn OneHotEncoder to the columns containing categorical data. |
| Large range of data in dataset that could cause bias in model. | Apply the sklearn MinMaxScaler function to the data. |

Note. Created by author 7th March 2023.

3.3.3.4.2 Feature Selection

Once the data has been processed, the datasets will be reduced to only the most important features. This will ensure only relevant features are presented to the model, improving accuracy. The importance of each feature will be identified using the sklearn mutual\_info\_classif function to calculate the mutual information between the feature and the target variable, in this case the label indicating whether the traffic is anomalous. Thereafter, importance will be ranked and plotted on a graph. From these the top 12 features will be selected.

3.3.3.4.3 Test and Training Split

Training and testing a model requires different data. Traditionally, functions such as test\_train\_split from the sklearn library are used to split the dataset randomly into two sets. However, in the review of the papers by Heine, Laue and Kleiner (2020) and Ring et al. (2019), the importance of not splitting data randomly, to keep time periods together to allow patterns to be identified, was highlighted. Therefore, the data must be split in a less traditional manner, by assigning specific segments of data to either the test or train set whilst maintaining a representative ratio of anomalous traffic, creating a larger training than test set and making sure the sets are the same size for both datasets. Moreover, the GAN, Autoencoder and Ensemble require only normal traffic for training and fitting. Therefore another set must be created containing only the normal traffic from the training set, however just a subset so that the number of samples is the same for each dataset. Further detail on the sets of data can be found in Appendix C.

#### Model Development

To build the models, the PyOD and sklearn python libraries will be used. As explained by Rashid and Miri (2021), PyOD is dedicated to outlier detection in multivariate data. Providing functions for implementing all the techniques required apart from Agglomerative clustering, which will be implemented through the sklearn library. For each of the models, the majority of the default values will be used, to keep the comparison fair, else it would be difficult to ensure similar parameter tuning has been applied to each model. However, there are a few parameters which must be defined; for the models to work and to keep the training fair. These are shown in the Table 9 below, alongside the reasoning for their values.

**Table 9** *Model Parameter Justification*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Parameter | Value | Reasoning |
| Agglomerative  Clustering | n\_clusters | 2 | The data is to be separated into two clusters; normal and anomalous. |
| AnoGAN | epochs | 100 | Both models train over a number of epochs therefore setting both to 100 keeps the comparison fair. |
| Autoencoder | epochs | 100 |
| hidden\_neurons | [12/4,12/8,12/4] | The number of hidden neurons should be relative to the input shape. Therefore, the same pattern used successfully by Le & Zincir-Heywood (2021) will be used. |

Note. Created by author 23rd February 2023.

## Evaluation Method

### Introduction

To meet the aim of the project, the selected techniques must be evaluated to measure their effectiveness. With effectiveness having been defined as a measure of a techniques ability to identify potential anomalies in user and entity behaviour. In the conclusion of the literature review, the metrics chosen to evaluate the methods, based on the findings of the review, were identified. The evaluation method will define a process to apply these metrics to evaluate the effectiveness of the models.

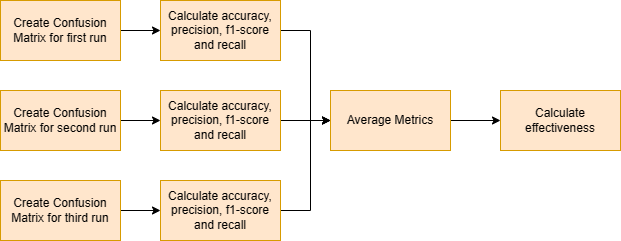
### Method

To evaluate the models, a confusion matrix will be created from which FPR, precision, accuracy, recall and f1-score will be calculated. Moreover, the models will be trained and tested three times to allow an overall average to be calculated for each metric. Furthermore, the BR of the test set will be displayed alongside the results for completeness, which was highlighted to be important by Heine, Laue and Kleiner (2020). Finally, the measure of effectiveness will be calculated, by summing the accuracy, f1-score and the negation of FPR to produce one metric to represent the effectiveness. The calculation is shown below.

𝐸𝑓𝑓𝑒𝑐𝑡𝑖𝑣𝑒𝑛𝑒𝑠𝑠 = 𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦 +𝐹1𝑆𝑐𝑜𝑟𝑒 −𝐹𝑃𝑅

As explained by Elmasry, Akbulut and Zaim (2019), f1-score is the harmonic mean of precision and recall. Therefore, it will be used rather than using precision and recall individually. Figure 7 provides a flowchart summarising the evaluation method that will be followed.

**Figure 6** *Evaluation Method Flowchart*

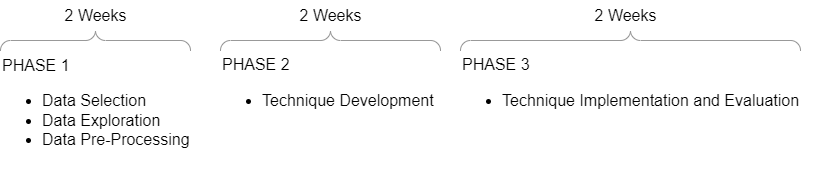


Note. Created by author 15th February 2023.

## Project Management Method

The initial project plan, shown earlier in the Gantt chart in Figure 1, designated four weeks for the development and testing of techniques and two thereafter for executing the experiment and evaluation. As the project is on track, this timeline will be followed. It will however be further broken into smaller phases covering the stages outlined above, with the breakdown shown in Figure 8.

**Figure 7** *Experiment and Evaluation Methods Time Management*



Note. Created by author 14th December 2022.

These time frames are a guide to ensure sufficient time is considered for each phase and to allow progress to be measured. Furthermore, a buffer has been incorporated into the project's timeline to account for any extra time required.

# Results and Discussion

## Introduction

Following the description of the experiment and evaluation methods, this section aims to document the experience of the experiment, including any difficulties that arose, changes to the proposed methods and comments on quality issues. Furthermore, the technical artefacts produced as part of the experiment and the results obtained through the evaluation will be presented. Thereafter, discussion will be provided on the analysis of the results and the critical evaluation of each technique. Concluding with the most effective technique being identified.

## Experiment Experience

Overall, the evaluation method was followed in full with no changes to the proposed activities. The experiment method was also followed in full for the most part, however, using the Kaggle notebook presented limitations that caused some of the steps to be adapted. Due to the limited processing power provided by the notebook, it was unable to pre-process all the data at once, meaning the data had to be processed in batches. Additionally, it was unable to handle the mutual information calculation for the complete datasets, meaning it had to be calculated for batches with the values combined thereafter. Furthermore, for the project management method, the time scales proposed for each phase of the experiment, were not followed in full. Due to the data preprocessing and technique development phases taking longer than expected due to unfamiliarity with the process. Leading to the required rigour and, processing time and power, being underestimated.

Furthermore, the steps proposed to preserve validity and reliability and minimised bias and error were implemented as planned. Additionally, the effect of steps such as using two datasets to reduce representation bias and using a selection of metrics to provide a transparent view of results, could be seen in the results. For many of the techniques, there was a large variance between the metrics produced on each dataset. If only one dataset had been used this would not have been observed leading to conclusions being drawn on less representative results. Moreover, it could be seen that some metrics, such as accuracy, were high whilst others, such as precision, were not. Suggesting that if a singular metric had been used the validity of the experiment could have been compromised as the metric would not have shown the complete performance of the technique.

## Technical Artefacts

One of the technical artefacts produced during the experiment was the code used for preprocessing each dataset. This code, included in Appendix A, implemented the steps required for pre-processing, discovered in the data exploration phase, through the activities outlined in the experiment method. It utilised the python panda’s library to filter out unwanted values and reformat the data. Additionally, for both datasets using the sklearn MinMaxScaler to normalise the data. Moreover, for the UNSW-NB15 dataset some extra steps were required, including one-hot encoding and removing unique identifiers. Furthermore, as previously mentioned the datasets were processed in batches due to the limited processing power of the Kaggle notebook.

Another technical artefact produced was the code, found in Appendix B, used for calculating the mutual information of features in the dataset. This code implemented the process outlined in the feature selection section of the data pre-processing phase of the experiment method. Allowing, graphs of mutual information of features to be plotted and the most important features to be selected. The graphs are included in Appendix B.

Moreover, code to implement each model was also produced. This code, found in Appendix D, utilised the python PyOD and sklearn libraries, as specified in the experiment method model development section, to create each model. Furthermore, code to evaluate the models was produced. This code, displayed in Appendix E, implements the process outlined in the evaluation method to calculate the required metrics, accuracy, precision, recall, f1-score, FPR and effectiveness.

## Results

This section will present the results of the experiment. These will be referred back to for analysis and critical evaluation in the discussion section that follows.

Table 11 shows the average metrics produced for each technique, for each dataset. The individual results for each of the three iterations of evaluations for each model, before being averaged, can be found in Appendix F. All metrics other than FRP and Effectiveness were rounded to two decimal places to allow two significant figures to be displayed where possible. FPR and Effectiveness were rounded to three decimal places as the range of the values was close to 0.1, therefore the extra digit proved informative. The results show a great variance between the metrics produced for each technique on each dataset, with only the AnoGAN performing consistently across both datasets. Agglomerative Clustering had the highest effectiveness for the CICIDS2017 dataset but the lowest on the UNSW-NB15 dataset. Similarly, the Ensemble proved most effective for the UNSW-NB15 dataset but proved less effective on the CICIDS2017 dataset. **Table 10** *Average Metrics and Effectiveness for each Model*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Dataset | Accuracy  (2dp) | Precision  (2dp) | Recall  (2dp) | F1Score  (2dp) | FPR  (3dp) | Effectiveness  (3dp) |
| Autoencoder | CICIDS  2017 | 0.84 | 0.50 | 0.44 | 0.47 | 0.086 | 1.224 |
| UNSW-  NB15 | 0.92 | 0.73 | 1.00 | 0.84 | 0.109 | 1.651 |
| OSVM and IF  Average  Ensemble | CICIDS  2017 | 0.80 | 0.53 | 0.53 | 0.53 | 0.130 | 0.930 |
| UNSW-  NB15 | 0.96 | 0.90 | 0.90 | 0.90 | 0.028 | 1.792 |
| AnoGAN | CICIDS  2017 | 0.80 | 0.54 | 0.56 | 0.55 | 0.131 | 1.219 |
| UNSW-  NB15 | 0.89 | 0.33 | 0.78 | 0.46 | 0.103 | 1.247 |
| Agglomerative  Clustering | CICIDS  2017 | 0.87 | 0.81 | 0.53 | 0.64 | 0.032 | 1.478 |
| UNSW-  NB15 | 0.02 | 0.00 | 0.00 | 0.00 | 0.974 | -0.972 |

Note. Created by author 1st March 2023.

Furthermore, following the recommendation made by Heine, Laue and Kleiner (2020), the BR of each test set is included below in Table 12. Including the BR allows the results to be interpreted relative to the datasets used and the influence on precision to be understood. Meaning that if someone wanted to compare these results with those of a separate experiment, using different datasets, the BR would help them do so.

**Table 11** *BR of the test set of each dataset*

|  |  |
| --- | --- |
| Dataset | BR of Test Set (3dp) |
| CICIDS2017 | 0.161 |
| UNSW-NB15 | 0.225 |

Note. Created by author 1st March 2023.

Additionally, Table 13 below displays the average effectiveness for each model across both datasets, sorted highest to lowest. Showing the Autoencoder to have been most effective overall. Table 12 *Average Effectiveness across Datasets*

|  |  |
| --- | --- |
| Model | Average Effectiveness (3dp) |
| Autoencoder | 1.438 |
| OSVM and IF Average Ensemble | 1.361 |
| AnoGAN | 1.233 |
| Agglomerative Clustering | 0.253 |

Note. Created by author 7th March 2023.

## Discussion

This projects aims to critically evaluate a selection of techniques for UEBA to identify the most effective techniques for analysing user and entity behaviour, specifically in enterprise network traffic. With effectiveness having been defined as a technique’s ability to profile normal behaviour and allow meaningful deviations to be identified. Calculated through combining metrics, accuracy, F1-score and FPR, which measure a technique’s ability to identify meaningful deviations from the normal.

### Identification of Most Effective Technique

In light of this, from the results shown in Table 13, the Autoencoder was observed to be most effective for analysing user and entity behaviour in enterprise traffic. Achieving the highest average effectiveness score across both datasets, with a score of 1.438. The OSVM and IF Average Ensemble followed closely, with an effectiveness score of 1.361. Emulating similar results to the experiment performed by Le and Zincir-Heywood (2020) whereby the Autoencoder performed on par with the Ensembles. The AnoGAN proved third most effective and showed the most consistent metrics across both datasets. Agglomerative Clustering achieved a slightly higher than average effectiveness score with the CICIDS207 dataset however proved completely ineffective with the UNSW-NB15 dataset, causing its average effectiveness score to be the lowest. This discrepancy between the performances on different datasets was not unique to Agglomerative Clustering.

### Effect of Dataset

When looking at the individual results per dataset in Table 11, the IF and OSVM Average Ensemble applied to the UNSW-NB15 dataset achieved the highest effectiveness score, 1.792. Achieving higher than any other technique in every metric calculated. Yet when applied to the CICIDS2017 dataset the performance was poorer, with an effectiveness score of only 0.930. Moreover, for the CICIDS2017 dataset, Agglomerative Clustering proved the most effective technique with a score of 1.478, however when applied to the UNSW-NB15 dataset the technique proved completely ineffective achieving the lowest recorded effectiveness, -0.972. A significant discrepancy between metrics produced for each technique on each dataset could be seen for all apart from the AnoGAN. This discrepancy can also be observed in experiments conducted in literature. In that carried out by Le and Zincir-Heywood (2020), large variance could be seen between the detection rates of algorithms when applied to different datasets. For example, at a 1% investigation budget the detection rate for the Autoencoder on the LANL dataset was 20.26% however on the TWOS dataset it was only 2.63%.

This variance places further importance on the recommendations made by Heine, Laue and Kleiner (2020), Fernandes et al. (2018) and Ring et al. (2019) that datasets should resemble the final deployment scenario as closely as possible, as performance varies significantly between datasets. Therefore, techniques that prove effective on one dataset maybe be ineffective on another, such as the Agglomerative Clustering technique in this experiment.

### Transparency of Metrics

Additionally, the concern expressed by Heine, Laue and Kleiner (2020) regarding the way singular metrics can skew the perception of a technique’s performance, can also be seen in the results. For example, all the techniques achieved an accuracy of at least 0.80 on the CICIDS2017 dataset however the recall of each was poor, with them all below 0.56. Showing the importance of using a selection of metrics to show the complete and representative performance of a technique.

## Conclusion

To conclude, the critical evaluation of the techniques showed the Autoencoder to be the most effective technique for analysing user and entity behaviour in enterprise network traffic, with the IF and OSVM Average Ensemble following closely behind. The analysis also showed the significant impact of the dataset on the technique’s performance, with the AnoGAN being the only technique that performed consistently across both datasets. Highlighting the importance of the datasets used to evaluate the techniques closely resembling the final deployment scenario, for the metrics to give a true representation of how the techniques will perform in the real world, supporting the findings of the literature review.

# Conclusion

## Introduction

Following the completion of the experiment, this section aims to summarise the findings of the report and draw together the conclusions and outcomes of the different phases of the project. Mapping them to the aims and objectives set out at the beginning of the project. Moreover, the limitations of the project will be acknowledged and areas of future work identified. Furthermore, self-appraisal of the project will be included.

## Project Overview

UEBA solutions derive baseline profiles of user and entity behaviour from which they flag deviations as potential anomalies. In literature, many different solutions have been presented for UEBA analysis, with a movement towards AI-based techniques being observed. Particular success was observed with Autoencoders, Ensembles and GANs. However little comparison existed between categories of algorithms such as Ensembles and GANs. Leading to an Autoencoder, Agglomerative Clustering, an IF and OSVM Average Ensemble and AnoGAN being chosen as the techniques to be evaluated in this project.

Evaluation of the techniques showed the Autoencoder to be most effective at identifying meaningful deviations in user and entity behaviour in enterprise network traffic, with the IF and OSVM Average Ensemble following closely behind. Additionally, the results of the experiment supported many of the key findings of the literature review, one of them being the effect of datasets on the metrics produced.

Despite a large volume of UEBA analytic solutions being published, less have been applied in practice. Believed to be due to a variety of factors, one being the effect of datasets on performance metrics produced. Leading to solutions that appeared high performing in literature not upholding that performance in practice. To address this, Heine, Laue and Kleiner (2020), Ring et al. (2019) and Fernandes et al. (2018) emphasise the importance of choosing datasets that closely resemble the final deployment scenario when evaluating techniques so that the metrics are representative of the analytic in a real-world scenario. This was further supported by the results of the experiment whereby the metrics produced for every technique apart from the AnoGAN differed significantly for each dataset.

Another factor, also supported by the results of the experiment, and believed to play into the discrepancy between the published work and practical applications of solutions, is the metrics used for evaluation. Heine, Laue and Kleiner (2020) and Axelsson (2000) both highlighted the importance of metrics providing a transparent evaluation, revealing that poor performance can be hidden behind singular high-performing metrics. The results of the experiment also supported this finding, showing some metrics remain high when others are very low.

## Project Evaluation

At the beginning of the project, the aim was set alongside a range of objectives to be fulfilled to achieve the aim. This section looks to identify the extent to which each of these has been met as well as to providing answers to the research questions.

### Aim

The Aim of the project was to critically evaluate a selection of techniques for UEBA to identify the most effective technique for analysing user and entity behaviour, specifically in enterprise network traffic. This was achieved through the implementation of different methods to evaluate each technique. Allowing the Autoencoder to be identified as the most effective technique for the datasets used, and out of the techniques evaluated. However, a great variance could be seen between effectiveness and the datasets used. Meaning the effectiveness of a technique appeared subjective to the dataset on which it was applied. Therefore, the aim of the experiment was successfully met, as a range of techniques were critically evaluated to identify the most effective. However, the variance between the performances on different datasets highlights that careful consideration must be applied when transposing the results to different scenarios. Highlighting, areas for future development. Furthermore, limitations of the experiment must be acknowledged.

### Objectives

Objective 1 was to review a range of contemporary literature to inform the methodology and analysis. The literature review presented in this report reviewed a range of literature on relevant topics and provided the theoretical underpinning of the project. Informing decisions throughout the project, such as the selection of datasets used, the techniques evaluated and the evaluation metrics utilised. Furthermore, through the review, recommendations for preserving validity and reliability, and, reducing bias and error, such as using multiple datasets and using a variety of evaluation metrics, were identified. Additionally, the findings of the review provided support for conclusions drawn from the results.

Objective 2 was to select techniques based on the findings of the literature review. Emerging and successful techniques for UEBA were identified during the literature review. Furthermore, a gap between the comparisons of techniques from different categories in literature was also highlighted. Justifying the selection of the techniques that were evaluated through evidence of their past success, and, the way the selection provided a comparison between techniques of different categories.

Objective 3 was to create and develop methods based on the findings of the literature review. As previously highlighted in the review of objective 1, the findings of the literature review greatly influenced the creation of the methods. Further seen in the inspiration provided by key pieces of literature, such as that by Le and Zincir-Heywood (2020), in the evaluation and experiment method. Moreover, the success of the methods can be observed by the way only limited changes were made during their implementation, most of which were due to inexperience, not the methods.

Objective 4 was to evaluate the techniques and gather performance metrics. Through the successful implementation of the evaluation method, performance metrics were gathered for each technique. Furthermore, steps incorporated to consider validity, reliability, bias and error proved effective and their impact could be observed in the results.

Objective 5 was to critically evaluate the results and identify the most effective technique. Critical evaluation of the results allowed the Autoencoder to be identified as the most effective technique. However, it also showed the effect of the dataset on the results produced and the importance of using transparent metrics.

Objective 6 was to draw conclusions and lay out future recommendations. Bringing together the results of the experiment and the key findings of the literature review allowed a range of conclusions to be drawn. Additionally, from the report so far, some limitations and future recommendations have already been identified, however, these will be expanded on and added to in the upcoming Limitations and Future Works sections.

### Research Questions

Research Question 1

From the experiment conducted, the results showed that out of the techniques evaluated the Autoencoder was able to detect anomalies most effectively.

Research Question 2

Reviewing the paper by Heine, Laue and Kleiner (2020) revealed that multiple metrics should be used when evaluating a technique to ensure that poor performance is not hidden behind a singular metric. Moreover, Axelsson (2000) highlighted, BR should be displayed alongside the metrics so that they can be understood with respect to the dataset. From these findings, it was decided that the metrics, accuracy, precision, recall, f1-score and FPR displayed alongside the BR would be most appropriate for the evaluation of the techniques in this experiment.

Research Question 3

Heine, Laue and Kleiner (2020), Fernandes et al. (2018) and Ring et al. (2019) highlighted the importance of choosing datasets that were representative of the final deployment scenario. Moreover, Heine, Laue and Kleiner (2020) recommended using multiple datasets to avoid representation bias. In the case of this experiment, as the techniques were required to be evaluated, the datasets required labels so that the true values were known. Therefore, the CICIDS2017 and UNSW-NB15 datasets proved to be the most appropriate datasets for the evaluation of the techniques in this experiment.

## Limitations

Even though the aim and objectives of the project were successfully met, the project does have some limitations, introduced by a variety of decisions that had to be made to keep the project feasible.

One of these limitations was due to the project being limited to evaluating techniques on only network traffic. This meant the performance of the techniques when applied to other datasets for UEBA, such as logs could not be seen. Additionally, both the datasets used contained synthetic traffic which does not represent real traffic as accurately as datasets of real traffic. However, they were chosen as they were labelled meaning the techniques could be evaluated.

Additionally, another limitation arose from only evaluating four different techniques, and only one from each category. This was done to keep the experiment feasible however one technique cannot be representative of an entire category as other techniques within that category may be better performing.

Moreover, using Kaggle as an experiment environment presented another limitation. Kaggle was used as it provided a central place to access all the datasets and python libraries required at a cost suitable to the experiment. However, this limited the experiment to the processing power of the notebook. In turn, limiting, the number of epochs models could be trained on and, the size of the data, and, the number of features, that could be used. Therefore, if an environment with greater processing power were to have been used the results may have been improved. Furthermore, the limited processing power also meant the data had to be pre-processed, and mutual information calculated, in batches.

Lastly, another limitation of the experiment was that the models all had minimum parameter tuning applied. This was to ensure that the same level of parameter tuning was designated to each model to keep the comparison fair. However, if the models had been developed further, they may have been able to achieve higher performance.

## Future Work

From both the limitations and key findings of the project, a variety of areas for future work have been identified.

In relation to the limitations of the project, it would be insightful for the project to be repeated with more techniques being evaluated from each category. Allowing a better picture of the performance of techniques from each category, to be gleaned. This could be performed in a similar way to the experiment by Le and Zincir-Heywood (2020) where a selection of Ensembles were evaluated or when Sun et al. (2019) evaluated a range of GANs, but providing those evaluations for multiple categories of techniques, not just one. Contrastingly it could also prove beneficial to repeat the experiment with greater parameter tuning of each model to identify the maximum performance that could be achieved by the models, rather than just the basic performance.

Regarding the key learning points of the project, the impact of datasets on the evaluation of models stood out from both the findings of the literature review and the results of the experiment. Heine, Laue and Kleiner (2020) and Ring et al. (2019) both recommended using datasets that closely resemble real-world traffic and highlighted the discrepancy between the volume of analytics published and their uptake in practice. However, at the same time, Heine, Laue and Kleiner (2020) also stated that no perfect dataset existed. Therefore, interesting future work could exist around developing datasets that allow models to be evaluated in a way representative of their final deployment scenario.

## Self-Appraisal

The process of this project allowed valuable lessons to be learnt, and a variety of skills and knowledge to be developed. However, the project also presented a variety of challenges, strengths and weaknesses.

Setting out a clear project management plan was one strength of the project as it ensured the project remain organised and progressed at a suitable pace. However, the time required for some stages, specifically the data pre-processing stage was underestimated, taking a lot longer than expected. This was due to lack of experience of data science projects and therefore an underestimation of the rigour, time and processing power required. If the project were to be conducted again, the time allocation for the data pre-processing phase would be extended.

Another strength of the project was the transparent evaluation of the techniques which considered key lessons learnt from the literature review. Ensuring downfalls of past research were not repeated and that the results were valid. This was down to the thorough approach taken for the literature review which provided a strong theoretical underpinning for the project.

A weakness of the project was the lack of understanding of the algorithms within the techniques used. Understanding these better would have allowed for not only the basic performance of the models to be compared but also for them to be tuned to investigate if their performance could be improved further. Improving personal knowledge of ML, specifically, the mathematical underpinnings is an area for future personal development that could improve future work.

Overall, the project has provided valuable insight into both academic research projects and data science projects. Allowing important skills and knowledge to be developed which will support future research and work.

# References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . I. (2016). TensorFlow:

Large-Scale Machine Learning on Heterogeneous Distributed Systems. (pp. 265–283). USENIX Association. doi:10.5555/3026877.3026899

Andres, B., Dilruksha, J., & McCandless, E. (2020). Flow-Based and Packet-Based Intrusion Detetcion using BLSTM. *SMU Data Science Review, 3*. Retrieved 11 30, 2022, from https://scholar.smu.edu/datasciencereview/vol3/iss3/8

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. In A. T. Security. doi:10.1145/357830.357849

Babu, S. (2020). Detecting Anomalies in Users - An UEBA Approach. *In Proceedings of the International conference on Industrial engineering and Operations Management, Dubai, UAE, March 10-12, 2020*, (pp. 863–876). Retrieved 11 07, 2022

Babu, S., & R, R. (2021). Anomaly Detection using User Entity Behavior analytics and Data Visualisation. *2021 8th International conference on Computing for sustainable Global Development (INDIACom)*, (pp. 842-847). Retrieved 11 07, 2022

Bakhareva, N., Shukhman, A., Matveev, A., Polezhaev, P., Ushakov, Y., & Legashev, L. (2019).

Attack Detection in Enterprise Networks by Machine Learning Methods. *2019 International*

*Russian Automation Conference (RusAutoCon)*, (pp. 1-6). doi:10.1109/RUSAUTOCON.2019.8867696

Barbara, D., Couto, J., Jajodia, S., & Wu, N. (2001). ADAM: a testbed for exploring the use of data mining in intrusion detection. *ACM SIGMOD Record*, 15-24. doi:https://doi.org/10.1145/604264.604268

Beazley, C., Gadiyq, K., Ravi, R., Roden, D., Ye, B., Abraham, B., & E, D. (2019). Exploratory Data Analysis of a Unified Host and Network Dataset. *2019 Systems and Information Engineering Design Symposium (SIEDS)*, 1-5. doi:10.1109/SIEDS.2019.8735640

Calvo, A., Escrig, J., Guijarro, J., & Siddiqui, S. (2022). OpenUEBA - A systematic approach to learn behavioural patterns. *nvestigación En Ciberseguridad: Actas De Las VII Jornadas*

*Nacionales (JNIC 2022): 27-29 De Junio De 2022, Palacio Euskalduna, Bilbao*, (pp. 216–

219). Retrieved 11 07, 2022

Care, J., Furtado, P., & Predovich, B. (2022). *Market Guide for Insider Risk Management*

*Solutions.* Gartner. Retrieved 01 04, 2023, from

https://www.gartner.com/doc/reprints?id=1-29RPXPBT&ct=220418&st=sb

Datta, J., Dasgupta, R., Dasgupta, S., & Reddy, K. R. (2021). Real-time threat detection in UEBA using unsupervised learning algorithms. *2021 5th International Conference on Electronics,*

*Materials Engineering & Nano-Technology (IEMENTech).*

doi:https://doi.org/10.1109/iementech53263.2021.9614848

Denning, D. E. (1987). An Intrusion Detection Model. *IEEE Transactions on Software engineering*, 222-232. doi:https://doi.org/ 10.1109/TSE.1987.232894

Diop, A., Emad, N., Winter, T., & Hilia, M. (2019). Design of an Ensemble Learning Behavior Anomaly Detection Framework. *International Journal of Computer and Information Engineering*, 547-555. doi:doi.org/10.5281/zenodo.3566299

Elsmary, W., Akbulut, A., & Zaim, A. (2019). Empirical study no multiclass classification-based network intrusion detection. *Computational Intelligence*, 919-954.

doi:https://onlinelibrary.wiley.com/doi/full/10.1111/coin.12220?saml\_referrer

Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P., Kumar, J., Srivastava, J., & Dokas, P. (2004). *MINDS - Minnesota Intrusion Detection System*, 199-218.

Fernandes, G., Radrigues, J., Carvalho, L., Al-Muhtadi, J., & Proenca, M. (2018). A comprehensive survey on network anomaly detection. *Telecommunications Systems*, 447-489. doi:https://doi.org/10.1007/s11235-018-0475-8

Gartner. (2018). *Market Guide for User and Entity Behaviour Analytics.* Retrieved 12 12, 2022, from https://www.intelligentcio.com/me/wp-content/uploads/sites/12/2019/01/gartnermarket-guide-for-ueba-2018-analyst-report.pdf

Gates, C., & Taylor, C. (2006). Challenging the anomaly detection paradigm: a provocative discussion. *In Proceedings of the 2006 workshop on New security paradigms (NSPW ’06). Association dor computing Machinery, New York, NY, USA*, (pp. 21-29). doi:https://doi.org/10.1145/1278940.1278945

Heine, F., Laue, T., & Kleiner, C. (2020). On the Evaluation and Deployment of Machine Learning Approaches for Intrusion Detection. *2020 IEEE International Conference on Big Data (Big*

*Data)*, (pp. 4594-4603). doi:10.1109/BigData50022.2020.9378479

Jallad, K., Aljnidi, M., & Desouki, M. (2020). Anomaly detection optimization using big data and deep learning to reduce false-positive. *Journal of Big Data*, 1-12. doi:https://doi.org/10.1186/s40537-020-00346-1

Kent, A. (2016). Cyber security data sources for dynamic network research. In N. Adams, & N. Heard, *Dynamic Networks and Cyber-Security* (pp. 37-66). Retrieved 11 30, 2022

Khaliq, S., Abideen Tariq, Z. U., & Masood, A. (2020). Role of user and entity behavior analytics in detecting insider attacks. *2020 International Conference on Cyber Warfare and Security (ICCWS).* doi:https://doi.org/10.1109/iccws48432.2020.9292394

Lazarevic, A., Kumar, V., & Srivastava, J. (2005). Intrusion Detection: A Survey. *Managing Cyber Threats. Massive Computing, 5*, 19-78. doi: https://doi.org/10.1007/0-387-24230-9\_2

Le, D. C., & Zincir-Heywood, N. (2021). Anomaly Detection for Insider Threats Using

Unsupervised Ensembles. *IEEE Transactions on Network and Service Management*, 1152-1164. doi:10.1109/TNSM.2021.3071928

Legg, P. A., Buckley, O., Goldsmith, M., & Creese, S. (2017). Automated Insider Threat Detection System Using User and Role-Based Profile Assessment. *IEEE Systems Journal*, 503-512.

doi:10.1109/JSYST.2015.2438442

Liu, H. (2021). A insider threat detection system based on user and entity behavior analysis.

*Journal of Physics: Conference Series*. doi:https://doi.org/10.1088/17426596/1994/1/012021

Liu, L., De Vel, o., Han, Q.-L., Zhang, J., & Xiang, Y. (2018). Detecting and Preventing Cyber Insider Threats: A Survey. *IEEE Communications Survey & Tutorials*, 1397-1417. doi:10.1109/COMST.2018.2800740

Lukashin, A., Popov, M., Bolshakov, A., & Nikolashin, Y. (2019). Scalable Data Processing

Approach and anomaly detection method for user and entity behavior analytics platform.

*Intelligent Distributed Computing XIII*, 344–349. doi:https://doi.org/10.1007/978-3-03032258-8\_40

Mahoney, M., & Chan, P. (2003). Learning rules for anomaly detection of hostile network traffic.

*hird IEEE International Conference on Data Mining*, (pp. 601-604). doi:doi:

10.1109/ICDM.2003.1250987

Martin, A., Fernandez-Isabel, A., Martin de Diego, I., & Beltran, M. (2021). A survey for user behavior analysis based on machine learning techniques: current models and applications. *Applied Intelligence*, 6029-6055. doi:10.1007/s10489-020-02160-x

Mehrabi, N., Morstatter, f., Saxena, N., Lerman, K., & Galstyan, A. (2021). A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 1-35. doi:10.1145/3457607

Mehta, S., Kothuri, P., & Garcia, D. (2018). Anomaly Detection for Network Connection Logs. doi:https://doi.org/10.48550/arXiv.1812.01941

Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*, (pp. 1-6). doi:0.1109/MilCIS.2015.7348942

Nortier, B., Simoiu, C., Chuah, E., Francesco, P., Berrada, G., Hoitzing, H., . . . Daravariu, V.-A. (2018). *Developing data science tools for improving enterpirse cyber-security.* doi:https://doi.org/10.5281/zenodo.3558251

Ongsulee, P. (2017). Artificial Intelligence, machine learning and deep learning. *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, (pp. 1-6). doi:10.1109/ICTKE.2017.8259629

Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for. *International Journal of Engineering & Technology*, 478-482. Retrieved 01 11, 2023

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E.

(2011). Scikit-learn: Machine Learning in Python. *Journal of Machine learning Research*, 2825-2830. doi:10.5555/1953048.2078195

Pelletier, Z., & Abualkibash, M. (2020). Evaluating the CIC IDS-2017 Dataset Using Machine Learning Methods and Creating. *International Research Journal of Advanced Engineering and*, 187-191. Retrieved 01 11, 2023

Ponemon Institute. (2022). *2022 Cost Of Insider Threats: Global Report.* Retrieved 11 08, 2022, from https://www.proofpoint.com/uk/resources/threat-reports/cost-of-insider-threats

Rashid, F., & Miri, A. (2021). User and event behavior analytics on differentially private data for anomaly detection. *2021 7th IEEE 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).* doi:https://doi.org/10.1109/bigdatasecurityhpscids52275.2021.00025

Ring, M., Wunderlich, S., Sheuring, D., Landes, D., & Hotho, A. (2019). A survey of networkbased intrusion detection data sets. *Computers & Security*, 147-167. doi:10.1016/j.cose.2019.06.005.

Roy, B., & Cheung, H. (2018). A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, (pp. 1-6). doi:10.1109/ATNAC.2018.8615294

Sabuhi, M., Zhou, M., Bezemer, C.-P., & Musilek, P. (2021). Applications of Generative

Adversarial Network Anomaly Detection: A Systematic Literature Review. *IEEE Access*, 161003-161029. doi:10.1109/ACCESS.2021.3131949

Sadowski, G., Litan, a., Bussa, T., & Philips, T. (2018). *Market guide for User and Entity Behavior Analytics.* Retrieved 11 07, 2022, from https://www.intelligentcio.com/me/wpcontent/uploads/sites/12/2019/01/gartner-market-guide-for-ueba-2018-analyst-report.pdf

Savenkov, P., & Ivutin, A. (2020). Methods of Machine Learning in System Abnormal Behavior Detection. *International Conference on Swarm Intelligence*, (pp. 495–505). doi:https://doi.org/10.1007/978-3-030-53956-6\_45

Saxena, N., Hayes, E., Bertino, E., Ojo, P., Choo, K.-K. R., & Burnap, P. (2020). Impact and Key Challenges of Insider Threats on Organizations and Critical Businesses. *Electronics*, 1460. doi:https://doi.org/10.3390/electronics9091460

Sharfaldin, I., Lashkari, A., & Ghorbani, A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, (pp. 108-116). doi:10.5220/0006639801080116

Shashanka, M., Shen, M. Y., & Wang, J. (2016). User and entity behavior analytics for enterprise security. *2016 IEEE International Conference on Big Data (Big Data)*, (pp. 1867-1874). doi:https://doi.org/10.1109/bigdata.2016.7840805

Singh, M., Mehtre, B. M., & Sangeetha, S. (2019). User Behavior Profiling using Ensemble

Approach for Insider Threat Detection. *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, (pp. 1-8). doi:10.1109/ISBA.2019.877846

Singh, R., & Srivastav, G. (2021). Novel Framework for Anomaly Detection Using Machine

Learning Techniques on CIC-ISD2017 Dataset. *2021 International Conference on Technological Advacements and Innovations (ICTAI)*, 632-637. doi:10.1109/ICTAI53825.2021.9673238

Slipenchuk, P., & Epishkina, A. (2019). Practical User and Entity Behavior Analytics Methods for

Fraud Detection Systems in Online Banking: A Survey. *Biologically Inspired Cognitive Architectures 2019*, 83-93. doi:https://doi.org/10.1007/978-3-030-25719-4\_11

Sun, Y., Guo, L., Li, Y., Xu, L., & Wang, Y. (2019). Semi-supervised Deep Learning for Network Anomaly Detection. *Algorithms and Architectures for Parallel Processing*, 383-390. doi:10.1007/978-3-030-38961-1\_33

Tian, Z., Luo, C., Hu, H., Su, S., Sun, Y., & Shang, M. (2020). User and Entity Behavior Analysis under Urban Big Data. *ACM/IMS Trans. Data Sci.*, 19. Retrieved 11 07, 2022

Umer, M., Sher, M., & Bi, Y. (2017). Flow-based intrusion detection: Techniques and challenges. *Computers & Security*, 238-254. doi:https://doi.org/10.1016/j.cose.2017.05.009

Zhao, y., Nasru, ah, Z., & Li, Z. (2019). PyOD: A Python Toolbox for Scalable Outlier Detection.

*Journal of Machine Learning Research*, 1-7. doi:10.48550/ARXIV.1901.01588

# Appendices

## Appendix A: Data Pre-Processing

Link back to Data Pre-Processing Section

Link back to Technical Artefacts Section

Table 15 explains the pre-processing steps that had to be implemented for each dataset.

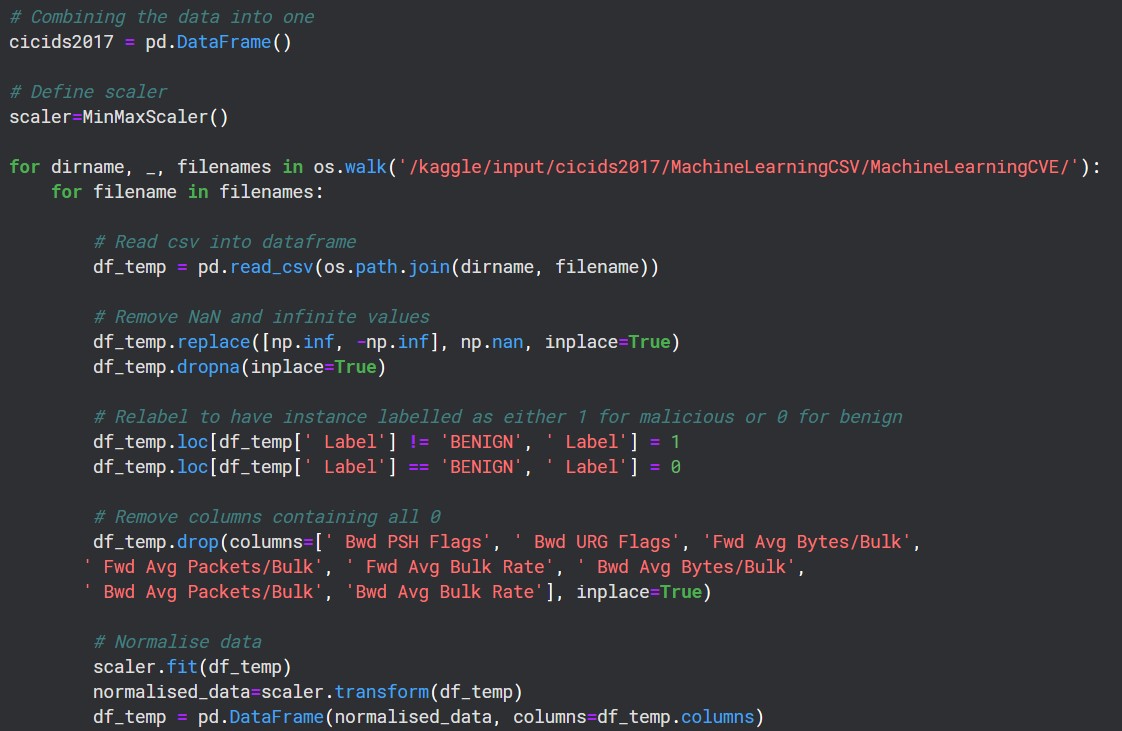
**Table 13** *Pre-processing steps to be applied to each dataset.*

|  |  |
| --- | --- |
| Dataset | Pre-Processing Steps |
| CICIDS2017 | For every csv:   1. Read the csv into a Panda’s DataFrame. 2. Remove NaN and infinite values. 3. Reformat the Labels to be 1 for anomalous and 0 for normal.   a. Remove columns containing all 0.   1. Normalise the data.   Thereafter concatenate the DataFrame onto one big DataFrame so as to have all the data together. |
| UNSW-  NB15 | For every csv:   1. Read the csv into a Panda’s DataFrame. 2. Drop the 'attack\_cat', 'ct\_flw\_http\_mthd', 'is\_ftp\_login', 'ct\_ftp\_cmd' as they are unrequired or full of 0’s. 3. Replace ‘-‘ values in the ‘service’ column with None so as to prevent errors. 4. In the ‘state’, ‘sport’, ‘dsport’ columns replaces the values listed below with NaN values.    1. ‘no’    2. ‘-‘    3. ‘0x000b’    4. ‘0x000c’    5. ‘0xc0a8’    6. ‘0x20205321’    7. 0xcc09’ 5. Drop all rows containing NaN values. 6. Remove the ‘srcip’ and ‘dstip’ columns as we are performing peer baselining. 7. One hot encode the columns containing categorical data; ‘proto’,   ‘service’ and ‘state’. |
|  | 8. Normalise the data.  Thereafter concatenate the DataFrame onto one big DataFrame to have all the data together. |

Note. Created by author 1st March 2023.

Figure 9 below shows the code used to pre-process the CICIDS2017 dataset.

**Figure 8** *Code for CICIDS2017 Pre-processing*



Note. Screenshot taken by author 1st March 2023.

Figure 10 below shows the code used to pre-process the UNSW-NB15 dataset.

**Figure 9** *Code for UNSW-NB15 Pre-Processing*



Note. Screenshot taken by author 1st March 2023.

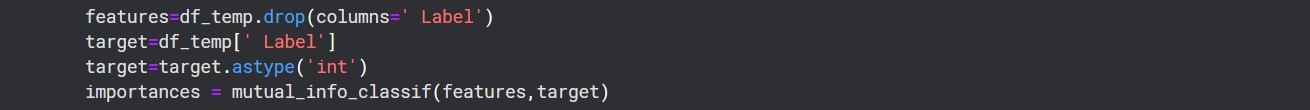
## Appendix B: Feature Selection

Link back to Technical Artefacts Section

The Kaggle notebook did not have the processing powered to apply the mutual information classifier to the whole dataset at once. Therefore, the mutual information for each feature was calculated for the data that made up each csv for each dataset and then the result of the mutual information from each were then combine and averaged to give the overall mutual information for each feature. Figures 11 shows the code used to calculate the mutual information for each csv.

This was included within the for loop that iterated through the csv’s, after the pre-processing was completed.

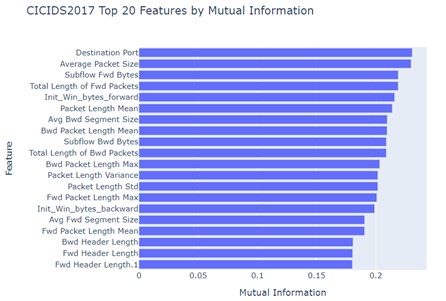
**Figure 10** *Code for calculating Mutual Information*



Note. Screenshot taken by author 1st March 2023.

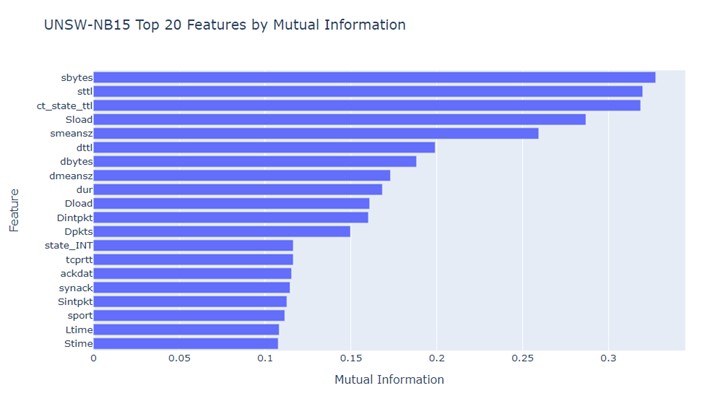
Figures 12 and 13 show the results of the feature selection process.

**Figure 11** *Top 20 Important Features of CICIDS2017*



Note. Created by author 11th January 2023.

**Figure 12** *Top 20 Important Features of UNSW-NB15*



Note. Created by author 11th January 2023.

## Appendix C: Testing and Training Data

Link back to Testing and Training Split Section

Table 16 below provides more detail on the data included in each testing and training sets.

**Table 14** *Training and Test Set Details*

|  |  |  |
| --- | --- | --- |
| Dataset | Testing/Training Set | Contents |
| CICIDS2017 | cicids2017\_X\_test | 15000 items from each of CVSs, 'Tuesday-  WorkingHours.pcap\_ISCX.csv', 'Thursday-  WorkingHours-Morning-  WebAttacks.pcap\_ISCX.csv' and 'Friday-  WorkingHours-Afternoon-  DDos.pcap\_ISCX.csv' with the items taken from index [10000:25000]. |
| cicids2017\_X\_train | Last 20000 items in each of the 'WednesdayworkingHours.pcap\_ISCX.csv', 'Thursday-WorkingHours-Afternoon-  Infilteration.pcap\_ISCX.csv' and 'Friday-  WorkingHours-Afternoon-  PortScan.pcap\_ISCX.csv' csv files. |
| cicids2017\_X\_train\_norm | The first 50000 normal traffic items in the cicids2017\_X\_train set. |
| cicids2017\_y\_train | The label column of the cicids2017\_X\_train set. |
| cicids2017\_y\_test | The label column of the cicids2017\_X\_test\_set. |
| UNSW-NB15 | unswnb15\_X\_test | 22500 items of each the 'UNSW-  NB15\_2.csv' and 'UNSW-NB15\_4.csv' csv files taken form index [-27500: -5000] |
| unswnb15\_X\_train | 30000 items of each the 'UNSW-  NB15\_1.csv' and 'UNSW-NB15\_3.csv' csv  files taken from index [10000:40000] |
| unswnb15\_y\_train | The first 50000 normal traffic items in the unswnb15\_X\_train set. |
|  | unswnb15\_y\_test | The label column of the unswnb15\_X\_train set. |
| unswnb15\_X\_train\_norm | The label column of the unswnb15\_X\_test\_set. |

Note. Created by author 1st March 2023.

Table 17 below provides more detail on the sizes and splits of data in the training and testing sets for each dataset.

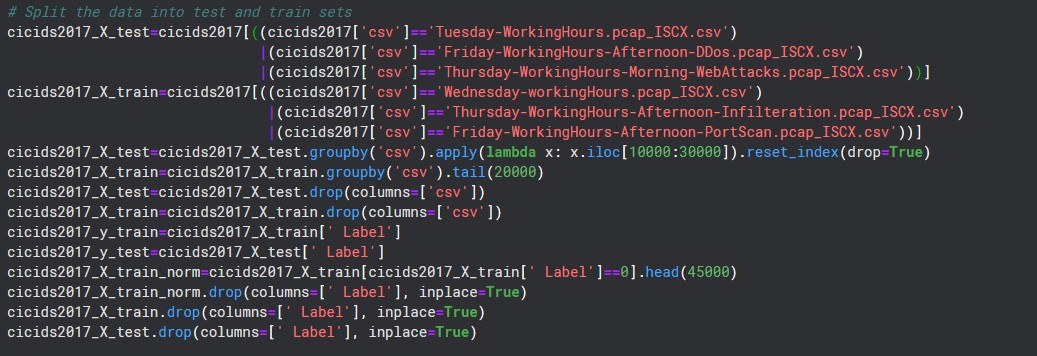
**Table 15** *Data sets size and split details*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | Testing/Training Set | Size | Split (Attack:  Normal) | BR (3dp) |
| CICIDS2017 | cicids2017\_X\_test | 45000 | 7234:37766 | 0.161 |
| cicids2017\_X\_train | 60000 | 10053:49947 | 0.168 |
| cicids2017\_X\_train\_norm | 45000 | 0:45000 | - |
| cicids2017\_y\_train | 60000 | - | - |
| cicids2017\_y\_test | 45000 | - | - |
| UNSW-NB15 | unswnb15\_X\_test | 45000 | 10117:34883 | 0.225 |
| unswnb15\_X\_train | 60000 | 11159:48841 | 0.186 |
| unswnb15\_X\_train\_norm | 45000 | 0:45000 | - |
| unswnb15\_y\_train | 60000 | - | - |
| unswnb15\_y\_test | 45000 | - | - |

Note. Created by author 1st March 2023.

14 below shows how the data was split for the CICIDS2017 dataset.

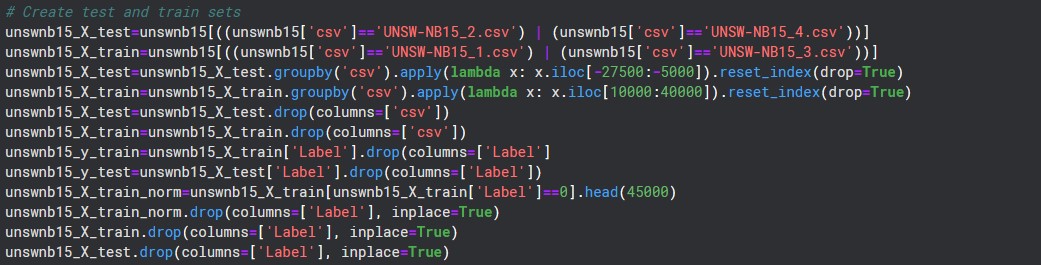
**13** *Code for splitting CICIDS2017 into its respective sets.*



Note. Screenshot taken by author 1st March 2023.

Figure 15 below shows how the data was split for the UNSW-NB15 dataset.

**Figure 14** *Code for splitting UNSW-NB15 into its respective sets.*



Note. Screenshot taken by author 1st March 2023.

## D: Model Definition

Link back to Technical Artefacts Section

Table 18 shows the functions to be used to implement each technique.

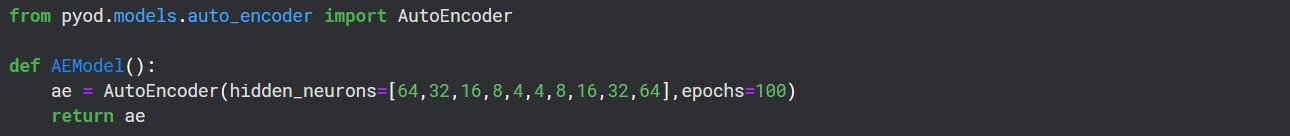
**Table 16** *Functions to be used to build each model.*

|  |  |
| --- | --- |
| Model | Function/s |
| AnoGAN | pyod.models.anogan.AnoGAN |
| Agglomerative Clustering | sklearn.cluster.AgglomerativeClustering |
| Autoencoder | pyod.models.auto\_encoder.AutoEncoder |
| IF-OSVM Ensemble | pyod.models.iforest.IForest pyod.models.OSVM.OSVM pyod.models.combination.average |

Note. Created by author 1st March 2023.

Figure 16 below shows the code used to define the Autoencoder.

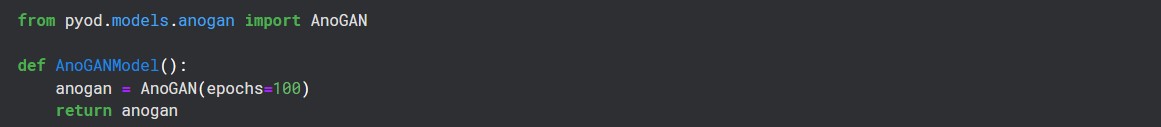
**Figure 15** C*ode for the Autoencoder.*



Note. Screenshot taken by author 1st March 2023.

Figure 17 below shows the code used to define the AnoGAN.

**Figure 16** C*ode for the AnoGAN.*



Note. Screenshot taken by author 1st March 2023.

18 below shows the code used to define the IF and OSVM Ensemble.

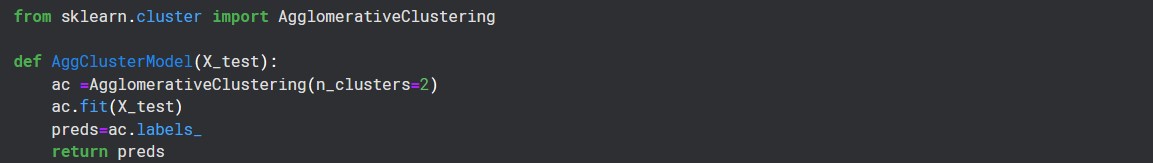
**17** *Code for the IF and OSVM Ensemble.*



Note. Screenshot taken by author 1st March 2023.

Figure 19 below shows the codes used to define the Agglomerative Clustering.

**Figure 18** C*ode for the Agglomerative Clustering.*



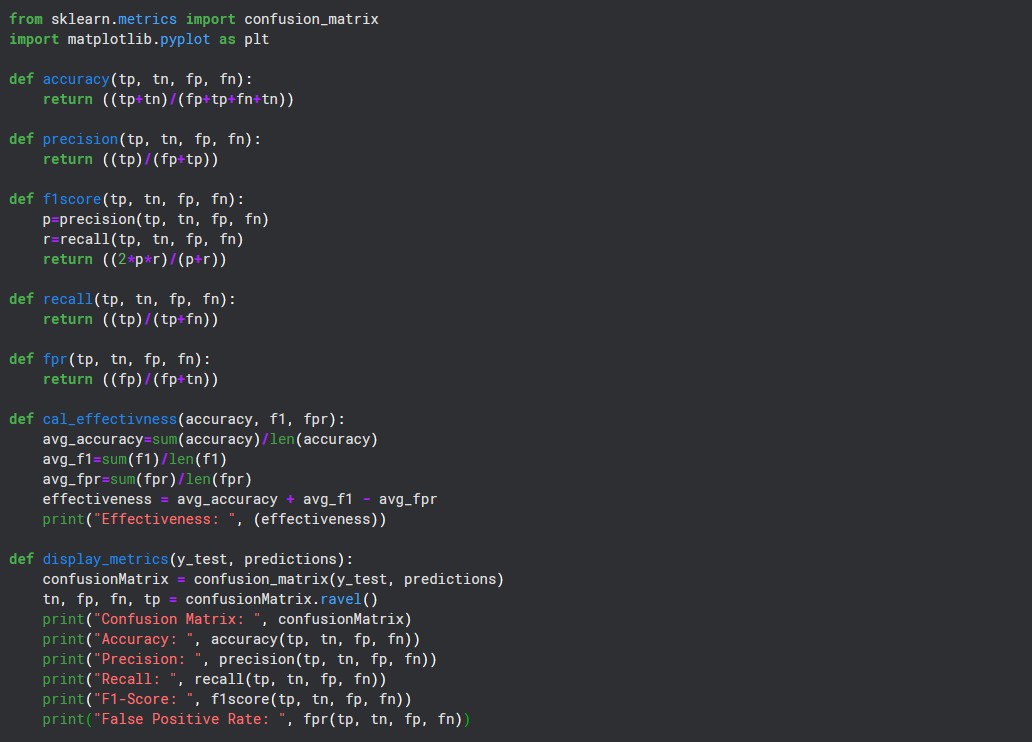
Note. Screenshot taken by author 1st March 2023.

## E: Evaluation Metrics

Link back to Technical Artefacts Section

Figure 20 below shows the code used to produce the evaluation metrics.

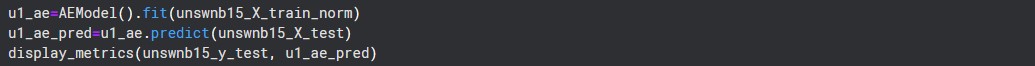
**Figure 19** *Code for the evaluation metrics.*



Note. Screenshot taken by author 1st March 2023.

Figure 21 below shows an example of the code used return the metrics from an Autoencoder model

**Figure 20** *Code for evaluating the Autoencoder.*



Note. Screenshot taken by author 1st March 2023.

22 shows an example of the code used to return the metrics after clustering **21** *Code for evaluating the Agglomerative Clustering.*



Note. Screenshot taken by author 1st March 2023.

Figure 23 shows the code used to return the metrics from an AnoGAN model **Figure 22** C*ode for evaluating the AnoGAN.*



Note. Screenshot taken by author 1st March 2023.

Figure 24 shows the code used to return the metrics from an Ensemble model **Figure 23** *Code for evaluating to IF and OSVM Ensemble.*



Note. Screenshot taken by author 1st March 2023.

## F: Individual Model Results

Link back to Results Section

Tables 19, 20, 21 and 22 show the complete results of the evaluations of each technique. The averaged results are presented in the results section.

**Table 17** *Results of the Evaluation of the Autoencoder.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Dataset | Accuracy  (2dp) | Precision  (2dp) | Recall  (2dp) | F1Score  (2dp) | FPR (3dp) |
| Autoencoder | CICIDS2  017 | 0.84 | 0.50 | 0.44 | 0.47 | 0.863 |
| 0.84 | 0.50 | 0.44 | 0.47 | 0.863 |
| 0.84 | 0.50 | 0.44 | 0.47 | 0.863 |
| UNSW-  NB15 | 0.92 | 0.73 | 1.00 | 0.84 | 0.109 |
| 0.92 | 0.73 | 1.00 | 0.84 | 0.109 |
| 0.92 | 0.73 | 1.00 | 0.84 | 0.109 |

Note. Created by author 1st March 2023.

**Table 18** *Results of the Evaluation of the IF and OSVM Ensemble.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Dataset | Accuracy  (2dp) | Precision  (2dp) | Recall  (2dp) | F1Score  (2dp) | FPR (3dp) |
| IF and OSVM  Ensemble | CICIDS2  017 | 0.80 | 0.53 | 0.53 | 0.53 | 0.13 |
| 0.80 | 0.53 | 0.53 | 0.53 | 0.13 |
| 0.80 | 0.53 | 0.53 | 0.53 | 0.13 |
| UNSW-  NB15 | 0.96 | 0.90 | 0.90 | 0.90 | 0.028 |
| 0.96 | 0.90 | 0.90 | 0.90 | 0.028 |
| 0.96 | 0.90 | 0.90 | 0.90 | 0.028 |

Note. Created by author 1st March 2023.

**Table 19** *Results of the Evaluation of the AnoGAN.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Dataset | Accuracy  (2dp) | Precision  (2dp) | Recall  (2dp) | F1Score  (2dp) | FPR (3dp) |
| AnoGAN | CICIDS2  017 | 0.80 | 0.54 | 0.55 | 0.55 | 0.131 |
| 0.81 | 0.55 | 0.57 | 0.56 | 0.130 |
| 0.80 | 0.54 | 0.56 | 0.55 | 0.131 |
| UNSW-  NB15 | 0.84 | 0.19 | 0.56 | 0.28 | 0.146 |
| 0.92 | 0.41 | 0.91 | 0.56 | 0.082 |
| 0.92 | 0.39 | 0.86 | 0.54 | 0.082 |

Note. Created by author 1st March 2023.

**Table 20** *Results of the Evaluation of the Agglomerative Clustering.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Dataset | Accuracy  (2dp) | Precision  (2dp) | Recall  (2dp) | F1Score  (2dp) | FPR (3dp) |
| Agglomerative  Clustering | CICIDS2  017 | 0.87 | 0.81 | 0.53 | 0.64 | 0.032 |
| 0.87 | 0.81 | 0.53 | 0.64 | 0.032 |
| 0.87 | 0.81 | 0.53 | 0.64 | 0.032 |
| UNSW-  NB15 | 0.02 | 0.00 | 0.00 | 0.00 | 0.974 |
| 0.02 | 0.00 | 0.00 | 0.00 | 0.974 |
| 0.02 | 0.00 | 0.00 | 0.00 | 0.974 |

Note. Created by author 1st March 2023.