National College of Ireland

A novel Deep Q-Learning algorithm for anomaly detection

MSc Research Project
Cloud Computing

Ammar Yousuf Abrahani
Student ID: 23341696

School of Computing
National College of Ireland

Supervisor: Dr Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ammar Yousuf Abrahani
Student ID:	23341696
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Dr Giovani Estrada
Submission Due Date:	11/08/2025
Project Title:	A novel Deep Q-Learning algorithm for anomaly detection
Word Count:	8106
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	AMMAR YOUSUF ABRAHANI
Date:	13th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A novel Deep Q-Learning algorithm for anomaly detection

Ammar Yousuf Abrahani
23341696

Abstract

The detection of anomalies is essential to identify unexpected patterns that can indicate fraud, security threats, or system failures. Classical models often depend on predefined rules or labelled data and become ineffective in dynamic or imbalanced environments. This research addresses these limitations by comparing advanced machine learning approaches, Deep Autoencoders and Deep Q-Learning (DQL) with classical models such as Isolation Forest and Local Outlier Factor (LOF).

Experiments were conducted on synthetic datasets generated with scikit-learn and the Kaggle Credit Card Fraud dataset. Isolation Forest emerged as a stronger classical baseline, achieving a macro-average F1-score of 0.934 on structured data. The Deep Autoencoder achieved a macro-average F1-score of 0.922, outperforming wide and shallow variants, while the DQL model demonstrated progressive improvement across 20 episodes, converging at approximately 0.92. However, in the imbalanced Kaggle dataset, both deep and classical methods experienced performance degradation (Autoencoder: 0.49) reflecting challenges in generalisation under extreme imbalance.

Unlike previous surveys that conceptually review reinforcement learning for anomaly detection, this study provides empirical benchmarking of reinforcement learning against both classical and neural models under identical conditions. The findings highlight that while deep and reinforcement learning approaches outperform static methods on structured data, their robustness diminishes with atypical distributions. This underscores the need for hybrid methods, temporal modelling, and hyperparameter optimisation to enhance real-world applicability.

1 Introduction

Anomaly detection is the task of identifying patterns that deviate from expected behaviour. These anomalies are often indicative of critical activities, such as fraudulent activities, failures, or cybersecurity attacks. Classical anomaly detection methods, such as thresholding, statistical modelling, and distance-based techniques, often struggle in noisy, high-dimensional industrial environments. In contrast, deep learning models such as convolutional autoencoders and hybrid architectures are more robust and scalable in visual anomaly detection tasks (Li et al.; 2025). Moreover, they typically require labelled data and do not perform well in anomaly-rare or anomaly-highly-variable situations.

Contemporary machine learning techniques are more flexible. Unsupervised neural networks, called autoencoders, have been catching our interest because of the good performance in learning compressed data representations and identifying high reconstruction

errors as suspicious data. Related works such as simple, deep, or wide autoencoders share significantly different depth and/or layer arrangement,which could affect their ability to model complex data distributions (Qiu et al.; 2024).

Besides neural methods, reinforcement learning (RL) based methods have been a very promising alternative to detect anomalies in systems that provide little or delayed feedback. Model-free RL (e.g., deep Q-learning, DQL) allows an agent to learn the best policies by trial-and-error without requiring labelled training data. Due to its adaptability and learning from the interaction history, DQL is well suited to detect evolving anomalies. Recent studies show that deep neural networks can significantly enhance anomaly detection when appropriately tuned.

This work first aims to improve traditional baseline models. It has been changed to Robust Covariance after doing a comparison with the local outlier factor (LOF) of the isolation forest. Based on this, three types of autoencoder architectures, simple, deep, and wide, were trained and tested on synthetic data to enquire into the effect of architectural complexity on performance. Then, a DQN model is trained and tested throughout various episodes to test the learning curve.

Generalisability was assessed by testing on an unbalanced real-world data set, the Kaggle credit card fraud detection dataset[1]. The data set is challenging, with rare anomalies and an unbalanced class distribution. Metrics such as macro F1-score were employed to guarantee equivalent performance in both majority and minority classes.

In summary, in this paper we investigate and compare the performance of deep learning and reinforcement learning models in anomaly detection, with respect to traditional algorithms. The results are presented in terms of the superiority of deep autoencoders and DQL with respect to synthetic and real anomaly detection problems.

1.1 UN Sustainable Goals

Although technically focused, this work also contributes to global agendas such as the United Nations Sustainable Development Goals (UN SDGs). In supporting Sustainable Development Goal 9.4, this research has proposed intelligent anomaly detection systems that can increase the operational efficiency of digital infrastructure. Using Deep Q-Learning (DQL) and autoencoder-based models, anomaly detection systems can reduce unnecessary computing workloads, while also reducing false positive alarms and avoid needless energy consumption through inefficient monitoring. Thus, the solution can enable cleaner computation and improved resource management in cloud and data-geared environments to help achieve a more sustainable digital infrastructure.

1.2 Research Gap

Although anomaly detection has been extensively studied, most traditional methods are heavily based on predefined thresholds, offline learning, or manual optimisation strategies. These characteristics make them less scalable and not robust enough for dynamic or high-dimensional environments. Machine learning approaches have attempted to improve anomaly detection capabilities, but they often require large, labelled datasets and may not generalise effectively across varied operational settings.

Recent interest in Deep Reinforcement Learning (DRL), particularly Deep Q-Learning (DQL), has opened new avenues for building adaptive real-time detection frameworks.

[1] `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`

Recent research indicates a growing trend towards deep learning-based approaches, such as autoencoders, Long Short-Term Memory (LSTM) networks, and transformers, capable of capturing complex data sets' temporal and non-linear dependencies more effectively (Wang et al.; 2025). However, there is limited work focused on the application of DQL for general purpose anomaly detection without relying on domain-specific rules or handcrafted features. In particular, (Wang et al.; 2025) presented a DRL-based framework for cyberphysical systems, demonstrating promising results in adaptive detection without thresholding. However, its application outside of industrial contexts remains largely unexplored.

This research aims to bridge that gap by introducing a DQL-based anomaly detection framework that is benchmarked against classical and neural methods. Unlike previous work, this study focusses on synthetic and real datasets of general use to assess the adaptability, precision, and robustness of DQL in a broader context. A range of architectures will be developed to determine which is the best performing.

1.3 Research Question

Despite extensive work on anomaly detection, particularly using reinforcement learning (RL) methods, it remains unclear which approaches are most effective in handling real-time anomaly detection under imbalanced data conditions. To address this gap, the following research question is proposed:

- What is the most effective Deep Q-Learning (DQL) method and network architecture for real-time anomaly detection, as evaluated by macro-average F1-score, precision, and recall?

1.4 Research objectives

In order to accomplish the research question, several steps have to be performed:

1. Implement traditional baseline anomaly detection models such as Random Forest and Isolation Forest.
2. Develop deep Q-learning algorithms for anomaly detection.
3. Compare the models performance with performance metrics such as macro-average F1-score, precision, recall under different anomaly scenarios.
4. Evaluate the scalability, responsiveness, and real-time adaptation of the anomaly detection models in various workloads.

1.5 Outline

This report is organised as follows. Section 2 offers a comparative background study on traditional and reinforcement learning anomaly prediction approaches. Section 3 describes the approach taken for collecting, pre-processing and evaluating data. The architecture and learning model of the proposed system are presented in Section 4. Section 5 describes the implementation process and the tools used. Section 6, Experimental and Comparative Results, is dedicated to experimental results and comparative analysis. Section 7 concludes the report by summarising the main findings and suggesting directions for further research.

2 Related Work

The detection of anomalies is a long-standing research problem. Traditional anomaly detection techniques have limited performance with large and complex modern systems, mainly due to an increase in data variability and dimensionality. Recognising anomalies has been a long-standing problem for researchers. Likewise, a recent survey by has provided an excellent overview of deep learning-based approaches to anomaly detection and the relative advantages they have over classical rule-based systems in high-dimensional and changing data (Pang et al.; 2021). Classical rule-based or statistic methodology is generally unable to adapt in real-time to changing behaviours, which has prompted the introduction of machine learning and deep learning techniques. In this section, we discuss the literature and classify previous work in several major areas of literature relevant to our study: traditional detection of anomalies, unsupervised machine learning models, and reinforcement learning-based adaptive detection.

2.1 Traditional and Machine Learning-Based Anomaly Detection Approaches

Anomaly detection has been an essential tool in system monitoring, fraud detection, and cybersecurity. Statistical models, distance-based methods, as well as rule-based heuristics were the early approaches to anomaly detection. In a recent survey of traditional anomaly detection methods (Chandola et al.; 2009), the material was grouped into statistical techniques based on proximity, proximity, and clustering. Although lightweight and intuitive, these approaches struggle in high-dimensional and dynamic environments in which patterns evolve rapidly over time.

Isolation Forest (iForest) is one of the pioneering unsupervised learning-based anomaly detection methodologies. Compared to density-based algorithms, iForest explicitly separates anomalies rather than characterising normal objects. With randomly chosen features and split values, it is able to detect anomalies in fewer steps compared to standard instances. This is computationally efficient and particularly suitable for high-dimensional datasets (Liu et al.; 2008). However, due to its fixed shape model nature, iForest is unable to capture the temporal or sequential nature that is prevalent in recent production logs and event sequences.

Deep learning methods have emerged as a solution to the aforementioned problems with traditional approaches. In DeepLog (Du et al.; 2017), the authors presented an LSTM-based model for system log anomaly detection that can model temporal dependencies and sequence-level anomalies. This algorithm improved substantially over standard methods for logarithmic anomaly detection. Nevertheless, DeepLog is still based on supervised learning, and may need to be retrained in case of changing system behaviours.

Recent studies have focused on AI based models utilising deep reinforcement learning (DRL) to obtain the best anomaly detection policy in non-stationary environments. They apply machine learning-based methods for zero-day and context-based anomaly detection considering the performance requirement of dynamic, real-time, and self-managed security solutions.

2.2 Reinforcement Learning and Deep Q-Learning for Adaptive Detection

Reinforcement learning (RL) is a machine learning paradigm that allows agents to learn decision making through their interaction with a given environment, either rewarding or punishing them. Deep learning architectures, such as autoencoders, CNNs, and GANs, have markedly improved the effectiveness of industrial visual anomaly detection by offering greater robustness to noise and variations compared to traditional hand-engineered or statistical techniques. Other authors conducted a comprehensive survey of this domain, systematically reviewing more than 200 recent studies and organising them in supervised, semi-supervised, self-supervised, and unsupervised learning paradigms, while also discussing dominant data sets, evaluation metrics, and key challenges (Mao et al.; 2025).

Additionally, compared to supervised learning methods, which are supervised and rely on labelled data, RL does not need labelled data, a property that makes it suitable for the detection of anomalies in data where anomalies are hard or impossible to label. The dynamic agent learns to maximise the long-term cumulative reward, which can be correlated to identify and react to anomalous behaviour in a dynamic environment.

Q-Learning is one of the widely used RL approaches, a value-based agent that computes the expected value (or utility) of taking an action in a state. Nevertheless, the classical Q-Learning approach faces various drawbacks as an effective anomaly detector in high-dimensional state space common in the real world. To deal with this, Deep Q-Learning (DQL) combines Q-Learning with deep neural networks, which makes it possible to approximate Q-values using function approximation. This has the benefit of enabling the agents to generalise over very large, continuous, or partially observable environments.

In recent years, many papers have investigated the use of adaptive and data-driven methods to enhance detection systems against anomalous behaviours. For example, Yahaya designed an adaptive anomaly detection model for human activity monitoring that incorporates temporal and context information to improve detection accuracy. Their model is able to adapt to change in behaviour over time instead of relying on static rule-based systems. This method demonstrates that it is crucial to include data-driven model learning on the fly in applications where patterns change continuously and it is very difficult to predict the data stream (Yahaya et al.; 2021).

Similarly, a Kubernetes anomaly detection architecture based on AI has recently been proposed (Bhardwaj et al.; 2024). Their approach employs reinforcement learning for discovering contextual and zero-day anomalies in containerised applications, which highlights the significance of adaptive and self-reliant architectures, such as cloud native. Other studies, a systematic taxonomy of deep learning methods for multivariate time series anomaly detection with a discussion of limitations in classical methods, and further identify the important gap which RL-based models seek to address (Wang et al.; 2025).

Although they show great potential, such works have several limitations, including sparse reward, complicated state representation, and convergence difficulties. Our system extends the existing body of knowledge by designing a DQL-based framework that addresses these issues via a rewards structure and state representations. Our approach enables us to formulate our anomaly detection problem, using our system to learn an optimal detection policy that increases prediction accuracy, detects more efficiently, and improves real-world implementation.

2.3 Comparison of Techniques and Emerging Trends in Anomaly Detection

The recent development of anomaly detection has produced a variety of methods that use conventional machine learning, neural networks, and RL. Each has strengths and weaknesses, and being aware of trends can help guide decisions related to future research directions and real-world applications.

2.3.1 Isolation Forest (IF)

The isolation forest algorithm (IF) is widely used and continues to be used for new applications. IF, in combination with Random Forest, has been used to mitigate and detect cyber security threats, including distributed denial of service attacks (DDoS), malware, and backdoor attacks occurring in networks (Ghani et al.; 2025). While the ability to process extremely high-dimensional data quickly is a significant strength, IF does not model time or changing patterns, which limits its application to ever-evolving environments.

2.3.2 Local Outlier Factor (LoF)

The local outlier factor (LoF) is another widely accepted method. Modifications of the LoF have been published that have been developed for industry applications. LoF was modified using PSO-VMD techniques to develop a method to diagnose voltage discrepancies in lithium-ion batteries (Li et al.; 2024). The modification aimed to improve the noise sensitivity and the ability to discriminate among minor variations. The limitations of LoF remain in that it is highly sensitive to the choice of neighbourhood parameters and methods fail in extremely high-dimensional cases or sparse datasets.

2.3.3 Autoencoders (AE)

Autoencoders (AE) are promising neural network-based methods that can learn compressed representations and identify anomalous characteristics using reconstruction loss. Recent studies have improved their adaptability to impaired samples by extending AEs with additional image denoising modules Richter et al. In another study, autoencoders were used to find intrusions in networks using a representation based on deep neural learning (Beg and Ansari; 2024). Despite their ability to detect complex patterns and signals, they often require extensive hyperparameter optimisation and may be more prone to overfitting, particularly with very high-dimensional data sets (Li et al.; 2025).

2.3.4 Deep Q-Learning (DQL)

Deep Q-Learning (DQL) and other reinforcement learning-dependent algorithms are beginning to garner interest due to their ability to learn optimal policy for detecting anomalies based on environment dynamics. In edge computing systems, DQL was used to reduce latency and improve adaptability to traffic changes (Kumaran and Sasikala; 2023). Similarly, in a different domain, DQL has been used to train flight controllers for fixed-wing aircraft, demonstrating that models could learn optimal control strategies from scratch without supervision (Richter and Calix; 2023). These examples illustrate the capacity and flexibility of DQL in high-risk dynamically evolving systems; nonetheless, convergence issues and reward sparsity need to be addressed.

The ability to assess anomaly detection methods, through the use of the macro-average F1-score, is common practice across all models. However, there are caveats to assessing F1 scores between models, as demonstrated by a Bayesian hierarchical model that represents uncertainty in average F1 comparisons (Zhang et al.; 2015). This modelling perspective is important in a research context with decisions involving averages, but the confidence intervals and robustness of the results also matter.

In addition, scikit-learn is a reliable tool for quickly prototyping and benchmarking machine learning algorithms, as demonstrated in a false news detection study that emphasises the performance improvements of feature selection in model performance (Sikarwar et al.; 2024). This research used scikit-learn integration with LoF, IF, and AE not only to make it easy to evaluate and tune the models iteratively but also to allow comparisons to be made under the same experimental conditions.

These various studies illustrate a common theme: the field of anomaly detection is transitioning towards adaptive, robust, and contextual systems. The values of adaptive models such as AE and DQL are highly adaptable, while the more sophisticated classical models of LoF and IF are key starting points of baseline models. The future is to develop hybrid architectures, transferable policies and real-time learning / feedback, which fits our assessment purpose of proposing a self-adaptive DQL-based anomaly detection framework. These various studies illustrate a common theme: the field of anomaly detection is transitioning towards adaptive, robust, and contextual systems. The values of adaptive models such as AE and DQL are highly adaptable, while the more sophisticated classical models of LoF and IF are key starting points of baseline models. The future is to develop hybrid architectures, transferable policies and real-time learning / feedback, which fits our assessment purpose of proposing a self-adaptive DQL-based anomaly detection framework.

2.4 Summary

This review of the literature explores recent techniques in anomaly detection, highlighting the shift from traditional methods to AI-driven approaches. Improved versions of the Local Outlier Factor (LOF) have been used for fault diagnosis in power systems, addressing sensitivity and parameter limitations. Autoencoders have proven to be effective in detecting complex anomalies, particularly in network intrusion and noisy environments. Hybrid models that combine Isolation Forest and Random Forest enhance cybersecurity by improving detection accuracy.

RL, especially Deep Q-Learning and Double DQL, has been applied to real-time systems such as edge computing and flight control, offering adaptability and low-latency detection. Supervised learning methods, including those implemented in Scikit-learn, demonstrate the value of proper feature selection in text classification tasks like false news detection. Statistical models such as Bayesian hierarchical frameworks offer reliable evaluation of model performance.

Overall, the literature reflects a trend toward adaptive, robust, and scalable AI solutions for anomaly detection, with deep learning and ensemble techniques offering superior performance in diverse applications.

The closest work is (Arshad et al.; 2022). However, that work did not explore different DQL architectures. The present report compares three autoencoders: simple, deep, and wide. There is also an analysis of the impact of MLP in depth vs. width. Moreover, the authors did not explore the effect of epochs in the DQL (to find the exact point to stop

learning). Another important difference between the works is in the evaluation metrics. Although (Arshad et al.; 2022) used precision, here the reported macro-average F1 score is more appropriate for imbalanced data. Here we show a comparison of metrics, compared to (Arshad et al.; 2022). Two datasets were used, one synthetic and one real-life.

3 Methodology

In this research, the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology was employed to systematically prepare, train, and test an anomaly detection system using reinforcement learning and machine learning. CRISP-DM is a standard approach to a systematic and rigorous data mining process, where evaluation can be performed at each phase of the process from mapping the original problem to the final model. This research relied on two types of datasets: one synthetic dataset that used scikit-learn to generate the dataset and a real-world dataset (Kaggle's credit card fraud detection)[2]

The reason for the synthetic data set is the balance of control provided. But a real dataset also runs under the complexity of some realism. Importantly, our goal is to benchmark and compare multiple models, specifically: Deep Q-Learning (DQL), autoencoder (AE), isolation forest (IF) and local outlier factor (LOF), using preprocessing, evaluation criteria, and metrics. The models will be evaluated in terms of macro-average F1-score, recall, precision, and adaptability to unknown datasets. The discussion surrounding each phase of the CRISP-DM will appear in this section as it pertains to how we used that phase in the context of our anomaly detection project.

3.1 Business Understanding

The fundamental business problem explored in this research is the lack of flexibility and robustness of classical anomaly prediction systems when operating in contexts where the data are noisy, dynamic, or unstructured. The vast majority of classical systems are either threshold-based or require significantly extensive manual tuning, which typically end in higher false-positive rates, lower chances of detecting anomalies, or poorly generalised models across datasets. To that extent, the aim of this research is to propose an intelligent monitoring system that learns how to autonomously detect types of anomalies by learning how to adapt to the different behaviours depending on types of distributions or types of noise.

The primary business problem that this research addresses is that traditional anomaly prediction systems are not very flexible and robust in environments that present noisy, variable, or unstructured data. For the most part, traditional systems are threshold-based or require a lot of human tuning, both of which eventually have higher false positive rates, lower anomaly detection accuracy, and cannot generalise across datasets (Xu et al.; 2017). Therefore, this research will aim to propose an intelligent monitoring system that autonomously detects anomalies by learning to accommodate shifting data distributions, and noise.

This study specifically compares existing approaches with deep Q-learning and compares their results with benchmark approaches such as autoencoders, Isolation Forest, and LOF, to see if reinforcement learning would offer more flexibility than those approaches without labelled data. This can apply to many industries such as finance (e.g. fraud

[2] `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`

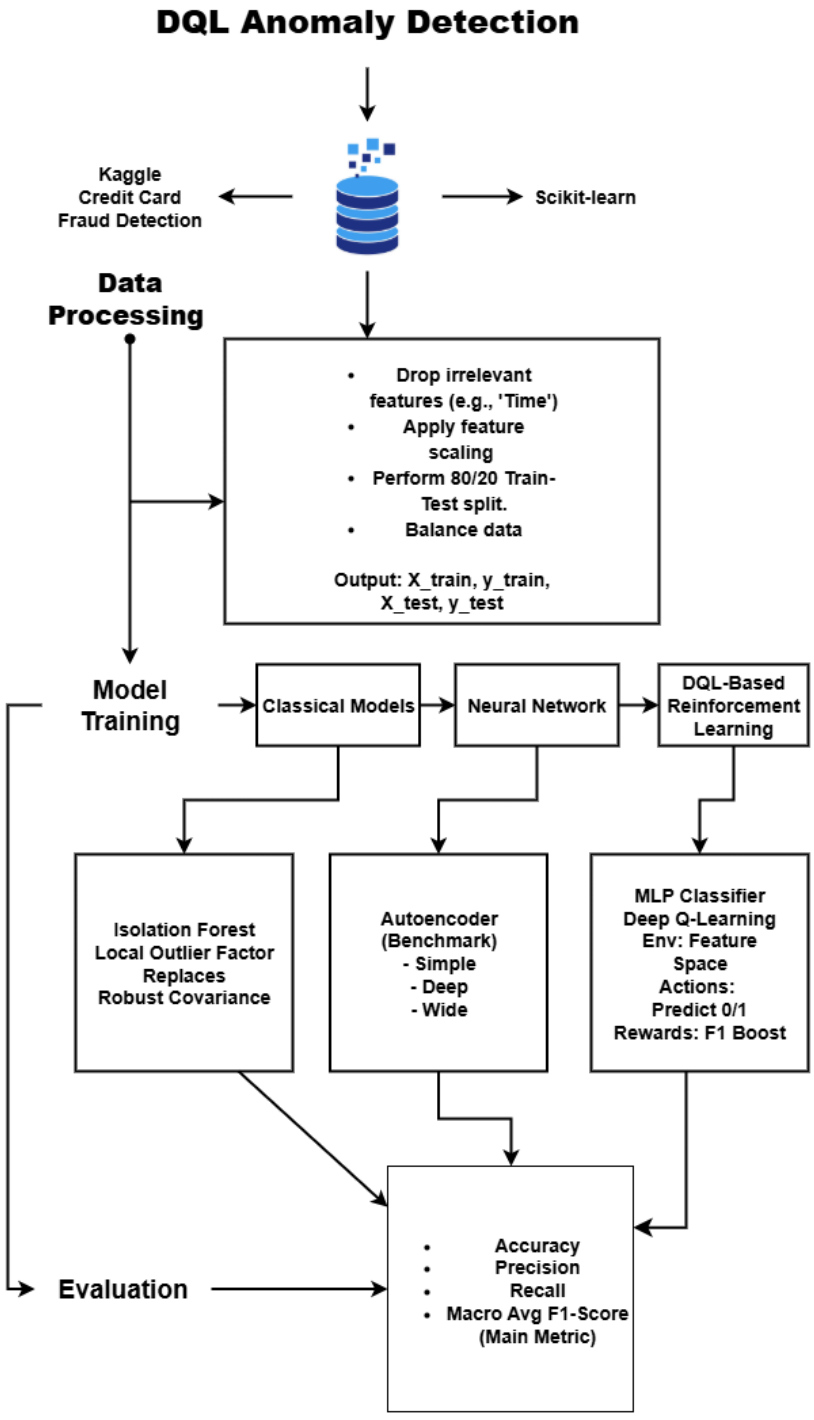

Figure 1: System Architecture

detection), health care (e.g. early risk identification), and cyber security (e.g. threats detection), since anomaly detection plays an important role (Arshad et al.; 2022).

Another driver to minimise false alarms, which can lead to alert fatigue and use unnecessary computational resources. Autoencoders have shown promise in avoiding the issue of false positives by learning input patterns and designating outliers with high reconstruction error (Chen et al.; 2023). Ultimately, the aim of this research is to create a detection system that can first demonstrate high precision and recall, and second, learn and improve on its own using feedback obtained from reinforcement learning. This model should help organisations build extensible and intelligent monitoring solutions.

3.2 Data Understanding

For data understanding, we investigated the properties of both the scikit-learn dataset and the Kaggle Credit Card Fraud Detection dataset. The datasets were generated using `make_blobs`, `make_moons`, and `make_classification` from Scikit datasets as examples of datasets that must adhere to complex, nonlinear decision boundaries due to some noise added to the dataset.

The synthetic datasets allow for labelled data, which allows us to test for detection success, while simulating anomalies through class imbalance and outlier features. Again, the synthetic data sets were beneficial in providing us with a baseline understanding of how each model functioned when both class and feature anomalies were clearly defined in relation to each classification boundary.

The Kaggle dataset includes extremely imbalanced data (fraud vs. non-fraud transactions), creating challenges for models to find and learn anomalies that are hardly different from many normal patterns. The initial understanding process required much effort into understanding the class distribution for the entire regression classification problem, while also incorporating charts evaluating correlation (i.e. heatmaps) and statistical summaries to ensure our strategies include proper preprocessing phase later. For example, the fraud class comprised only 0.17% of the total dataset, which introduces interesting challenges for all anomaly detection algorithms.

The data understanding phase involved visualisations through t-SNE and PCA to understand the spatial position of each of the classes and to determine if there were boundary specifications that the models could ultimately learn. In this case, we were also able to see in the datasets that we had analysed missing values, duplicates, and outliers, which we documented and modified accordingly.

3.3 Data Preparation

The data preparation involved an array of pre-processing methods to convert raw datasets to usable formats for training models in anomaly detection. Starting with the Kaggle dataset, we looked for missing values and found no missing values and normalised all features using Min-Max scaling to bring all values in the range [0, 1]. This normalisation was beneficial when stabilising the training process, but was statistically important, especially for neural networks like the autoencoder and DQL model. The class imbalance was corrected by performing stratified splits to maintain the same proportion of class in the train and test sets.

For the Scikit-learn synthetic dataset, we generate synthetic binary and multiclass datasets. The class imbalance was also intentionally created, and the scaling was applied

uniformly to each feature. These data sets were split 80/20 into train and test data sets calculated using fixed random seeds to ensure repeatability.

The training for autoencoder used only the majority (non-anomalous) class to determine the baseline pattern for reconstruction error on all other classes at inference time to determine if a point was anomalous. The isolation forest and the LOF were directly given the flattened vectors of the features.

The DQL model required significantly more effort to define the environment and learning dynamics than the other models. Each of the key aspects of the DQL model used for State 1, had to be defined. The state space is designed as flattened vectors (of feature values), and the action space was binary (0 for normal; 1 for anomaly). The main difference between DQL and the other models is that DQL consists of an agent learning on a trial-and-error basis, using interaction, which means the agent will have to build a reward function, set learning rate, discount factor, and design episodes and replay memory. The definition of an episode in this model could be defined as the actions or transactions that occurred. Each episode was defined by 100-200 different transactions (states) at different points within each transaction, along with how classification was done or is still being done based on properly classifying transactions. A reward was defined from positive points for right classifications (normal and anomalous) to negative points for misclassifying transactions (normal and anomalous); therefore, the DQL model required more planning but was more flexible to changing or adapting over time.

3.4 Modelling

The modelling stage consisted of developing four different models of anomaly detection; DQL, AE, IF, and LOF. Of these, DQL was the novel part of this research, and the other three models were used as classical baselines for comparison purposes. All models were evaluated and trained on the same Scikit-learn datasets to have a fair and consistent experimental basis.

The Deep Q-Learning agent is a model-free reinforcement learning approach in that it learns optimal policies through environmental interaction. In the realm of anomaly detection, the environment was modelled using time series sequences of individual system metrics as training input for the DQL agent. The time series sequences as state are the input when training the DQL agent. The action space was a binary decision, the state is anomalous or normal. The reward was intentionally designed to facilitate meaningful learning, with +1 given for correct classifications, -1 when the agent falsely predicted the actual classifications, and 0 for uncertain cases (when the agent was uncertain). This gives the agent sparse and structured signals to reward behaviour, maximising its performance in the long term.

The Q network was our Q-value approximant, which consisted of an input layer that accepts normalised state vectors, followed by two hidden layers with ReLUs and a final layer that predicts Q-values for actions. The model was trained using a standard Adam optimiser, and the MSE loss between predicted Q values and target Q values generated from the Bellman equation, as well as an epsilon-greedy strategy with a decaying epsilon throughout the training episodes, makes the training exploration-exploitation balance effective.

3.5 Iterative Improvement

The iterative nature of the CRISP-DM methodology is one of its greatest strengths because it encourages refinement based on interim findings. The modelling phase included multiple rounds of optimisation of the DQL agent (as well as baseline models) in terms of accuracy, generalisability, and robustness.

At first, the DQL agent performed today due to reward sparseness, early convergence, and overfitting. The DQL reward function has been revised to reflect the actual cost of making the wrong decision. For example, we determined that false negatives incur a higher penalty than false positives when their application is assessed in the real world. The agent epsilon decay was adjusted to increase exploratory actions early in training and increase exploitative actions later on. The memory replay buffer was implemented and used for experience sampling and allowed the agent to learn from a broader collection of past interactions.

For autoencoder models, the model configuration was equally important. The chosen configuration (deep model with the application of regularisation layers) performed better during the trial than the shallow architecture with regularisation and was less likely to overfit compared to the shallow model.

Classical models like Isolation Forest and LOF were optimised by tuning the various hyperparameters. For IF, the number of estimators and the contamination rate were tuned, while LOF was tuned for the number of neighbours and the contamination rate. Cross-validation was used wherever possible so that the model residuals could remain unbiased.

Visualisations such as anomaly score histograms, decision boundary plots, and Q-value heatmaps were very useful for inference and debugging of model behaviour, which in turn helped shape further tuning decisions and improve interpretability.

By going through the iterative improvement step, we saw that the final DQL agent performed very well when it came to adapting to a new anomaly, had relatively few false positives, and had high macro F1 scores. Moreover, this iterative approach was not just to improve model performance. Where operational reality is captured with imperfect information, the conditions are in flux, and models are developing and refining their adaptiveness.

4 Design Specification

4.1 Contribution

Although autoencoders for anomaly detection have been reported in the literature, no detailed study of the best-performing architectures has been published. Here, a comparison of three different autoencoder structures is presented, namely:

- Simple autoencoder (1–2 hidden layers).
- Deep autoencoder (multiple deep layers).
- Wide autoencoder (broader hidden layers with more units).

The evaluation was carried out using the macro-average F1-score, on two different datasets. In the remaining sections, we will see that a deep autoencoder could reach a

macro-average F1-score of ∼96%. A comprehensive experimental pipeline was designed to compare traditional, neural, and reinforcement learning approaches side by side.

The proposed hybrid anomaly detection model combines Deep Q-learning (DQL) with known unsupervised benchmark models including autoencoder, isolation forest, local outlier factor (LOF), and Multi-Layer Perceptron (MLP). The objective is to bring attention to the anomalies of different structure datasets by using classical unsupervised models, reinforcement learning, and neural networks in supervised / unsupervised methods. Two different data sets were used to test the framework: the Kaggle credit card fraud detection dataset[3] and Scikit-learn sample code[4] (e.g. blobs, moons, and random blobs with uniform noise).

4.2 Developed Artefact

The code is modularly structured so that the parts work in concert to perform end-to-end anomaly detection. The components include: Data Preprocessing Pipeline: Standardisation of features, processing of class imbalances and organising datasets for different model needs. Framework of models: Allows plug-and-play for DQL, Autoencoder (several architectures), Isolation Forest, LOF, and MLP.

4.3 Reward-Driven RL Environment

The environment is formulated on purpose for the DQL agent, where the state is represented with feature vectors, and the actions with the class label. Visualisation and Evaluation Suite: The suite contains modules to monitor model accuracy, macro-average F1-score and generate performance visualisations for episodes [5].

4.4 Architecture and Model Flow

The system has the following high-level control flow:

4.4.1 Data Input

Both real-world (Kaggle) and synthetic (Scikit-learn) data sets.

4.4.2 Model Training

Unsupervised Classical: Isolation Forest and LOF; Autoencoder (Unsupervised Neural Network); MLP (Supervised Neural Network); DQL Agent (Reinforcement Learning); and finally, Evaluation and Visualisation.

The agent operates in an environment which is an artificial data set that can learn from and interact with observation, performs action (normal / abnormal), and receives a reward (1 / -1).

[3] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[4] https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_anomaly_comparison.html

[5] https://github.com/ammarabrahani/deep-q-learning-anomaly-detection

4.5 A note on RL for anomaly detection

In stark contrast to standard unsupervised methods, such as Isolation Forest and LOF, which rely on the construction of tree splits or density estimates, the DQL leverages cumulative reward over episodes for updated anomaly classification. DQL generalises better on unseen patterns compared to a reconstruction loss-based method (autoencoder).

5 Implementation

The details of the implementation of the DQL-based anomaly detection technique are presented. The development process was modularised and provides insight into the preprocessing, model architecture combination, training strategies, and testing stages. All coding files were numbered by sequence of numbers for ease of reading. A complete README provides the details on how to set up the project, file execution order, and environment configuration[6]. Code snippets and screenshots are included in the Configuration Manual.

5.1 Data Preparation & Feature Engineering

The project was implemented using two main datasets: the Kaggle's Credit Card Fraud Detection dataset and the scikit-learn datasets. These data sets provided a variety of structures and distributions to train and test various anomaly detection approaches.

Preprocessing consisted of removing the column Time (which was not useful for learning the model) and then normalisation by StandardScaler. This data was divided into 80% training and 20% test sets. To train autoencoder-based anomaly detection, "normal" (ie non-fraudulent) instances were used in the training to match the non-fraudulent instances.

Furthermore, for performance comparison, the data set was restructured to maintain class imbalance, helping to evaluate the robustness of the model under realistic fraud detection conditions.

5.2 Training & Evaluation Process

For each category of model, the training protocols are as follows:

5.2.1 Autoencoder and MLP

The models were trained 100 epochs and used an early stopping criterion based on the validation loss. The reconstruction loss used during autoencoder training was the mean squared error. The F1 score is calculated after reconstruction of the anomaly detection based on threshold.

5.2.2 Classical Models (IF, LOF)

We trained the models with contamination rates = 0.15 in scikit-learn.

[6] `https://github.com/ammarabrahani/deep-q-learning-anomaly-detection`

5.2.3 Deep Q-Learning (DQL)

The agent was trained for 20 episodes. The macro F1 score is computed and visualised over episodes after each episode to demonstrate the improvement pattern. The new states have been added to the environment through replay memory, and through the ε -greedy strategy, exploration and exploitation have been balanced.

Macro-average f1 score, precision, recall, and accuracy were used to evaluate all models. The evaluation results were kept in tabulated views and the visualisations (that is, DQL F1 evolution and confusion matrices) were plotted using matplotlib.

6 Evaluation

This section presents the experimental setup, observations, and findings in multiple use cases. Each use case highlights a specific model or architecture used for anomaly detection in the context of this research. Evaluation is performed using the macro-average F1 score as a key metric.

6.1 Use Case 1: Selection of baseline

There are many classical algorithms for anomaly detection. The problem is which one should we use for the baseline. Therefore, the purpose of this use case is to identify a stronger classical baseline model for anomaly detection. It was observed that robust covariance model performed poorly on complex or non-Gaussian datasets, but a proper comparison was carried out to select the baseline. Two alternative unsupervised models, the isolation forest and the local outlier factor (LOF), were also included in the list.

A comparative evaluation was carried out using the isolation forest and the local outlier factor in data sets generated through the Scikit-learn library. These data sets represent varied shapes and density distributions, including circular blobs, clusters, and spirals.

Table 1: Comparison of Classical Anomaly Detection algorithms

Model	**Macro F1-Score**	**Behaviour**	**Selected**
Robust Covariance	∼0.40 (poor)	Fails on non-Gaussian data	✘
Local Outlier Factor	∼0.43	Inconsistent, fails on spirals	✘
Isolation Forest	**0.93**	Stable and effective	✓

Result: Isolation Forest consistently outperformed the other baseline algorithms in multiple scenarios, especially on complex data shapes such as spirals and multi-cluster distributions. Based on these results, the isolation forest was chosen as the baseline algorithm.

6.2 Use Case 2: Autoencoder (Simple / Deep / Wide) with Scikit-Learn Dataset

The goal of this experiment is to evaluate and compare the effectiveness of different Autoencoder architectures Simple, Deep, and Wide on a dataset generated using the

Scikit-learn library. These architectures were chosen to analyse the trade-offs between depth, width, and regularisation techniques in the context of anomaly detection.

Each variant of the Autoencoder model was trained individually on the same pre-processed dataset. The data set was standardised and the labels were retained only for evaluation purposes. The autoencoder models were implemented using Keras and evaluated using macro average F1-score and accuracy.

- **Simple Autoencoder:** A shallow network consisting of two dense layers, specifically Dense(32) followed by Dense(2). This minimal architecture captures only basic compression features and serves as the baseline configuration.
- **Deep Autoencoder:** A deeper architecture comprising six layers with Dropout and Batch Normalisation applied to mitigate overfitting and improve generalisation. Additional deep variants were tested by varying:
  - The number of hidden layers (e.g., 4, 6, 8)
  - Neuron counts per layer (e.g., 64, 32, 16)
  - Dropout rates between `0.2`
  - With and without Batch Normalisation

  These tests aimed to explore the effect of regularisation, depth, and network complexity on generalisation performance.
- **Wide Autoencoder:** A two-layered architecture designed with a high number of neurons per layer, emphasising width over depth. The implemented structure was

  `Dense(128)` $\rightarrow$ `Dense(128)` $\rightarrow$ `Dense(2)`

  These were aimed at understanding whether broader layers without added depth could improve anomaly detection.

Table 2: Performance Comparison of Autoencoder Architectures

Architecture	**Macro F1-Score (Autoencoder)**	**Accuracy**
Simple Autoencoder	0.489	0.87
Deep Autoencoder	**0.922**	**0.96**
Wide Autoencoder	0.521	0.89

Result: Among all three architectures, the Deep Autoencoder significantly outperformed the Simple and Wide variants, achieving a macro-average F1-score of 0.922 and an accuracy of 0.96. Figure 2 shows the best architecture found. This indicates that increased depth, along with regularisation techniques such as Dropout and Batch Normalisation, contributes substantially to the model's anomaly detection capabilities. The Simple Autoencoder, while efficient, struggled with learning complex representations, resulting in a lower F1-score. The wide autoencoder performed slightly better than the simple model, but failed to match the robustness of the deep architecture.

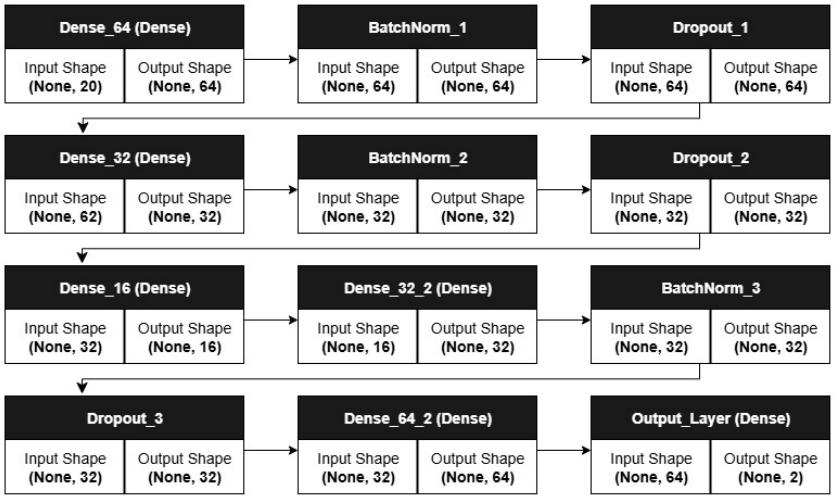

Figure 2: Deep neural network architecture used for anomaly detection. The model consists of a sequence of Dense, Batch Normalization, and Dropout layers arranged in a deep sequential configuration.

6.3 Use Case 3: Autoencoder with Kaggle Credit Card Fraud Dataset

Having found a good architecture for anomaly detection, we proceed to apply it to real data. Evaluation of the robustness and generalisation of the Deep Autoencoder architecture on a real-world imbalanced dataset.

The same Deep Autoencoder architecture was trained. A 90th percentile threshold (0.45151) was used for binary classification. The classification results are presented below:

Table 3: Classification Report at 90th Percentile Threshold (0.45151)

Class	Precision	Recall	F1-Score
Normal	1.00	0.90	0.95
Anomaly	0.02	0.91	0.03
Macro Avg	0.51	0.88	**0.49**

Result: Table 3, despite a high recall, the model exhibited very low precision for anomalies. This highlights the limitations of unsupervised learning on highly imbalanced datasets where false positives are more frequent. The corresponding macro-average F1-Score is 0.49, which is unacceptable for any practical application. High accuracy was observed (90%), but is misleading due to class imbalance; therefore, the macro-average F1 score offers a better evaluation metric in this context.

Summary of results: Table 4 lists all anomaly detection algorithms to better compare their performance in the real-life data set:

Model	Macro F1-Score
Deep Autoencoder	**0.922**
Isolation Forest	0.934
MLP Classifier	0.889
Local Outlier Factor	0.430

Table 4: Macro F1-Score Comparison Across All Models

6.4 Use case 4: parameter study of deep Q-Learning algorithm

A very important parameter in a DQL algorithm is the number of episodes. Little is known on how to select it. The results shown autoencoders and DQL are the most effective in the macro F1 score, particularly when properly architected. Their superior ability to model complex patterns in the data provides significant advantages in detecting anomalies compared to classical techniques. Therefore, a parameter study was carried out and presented in Figure 3.

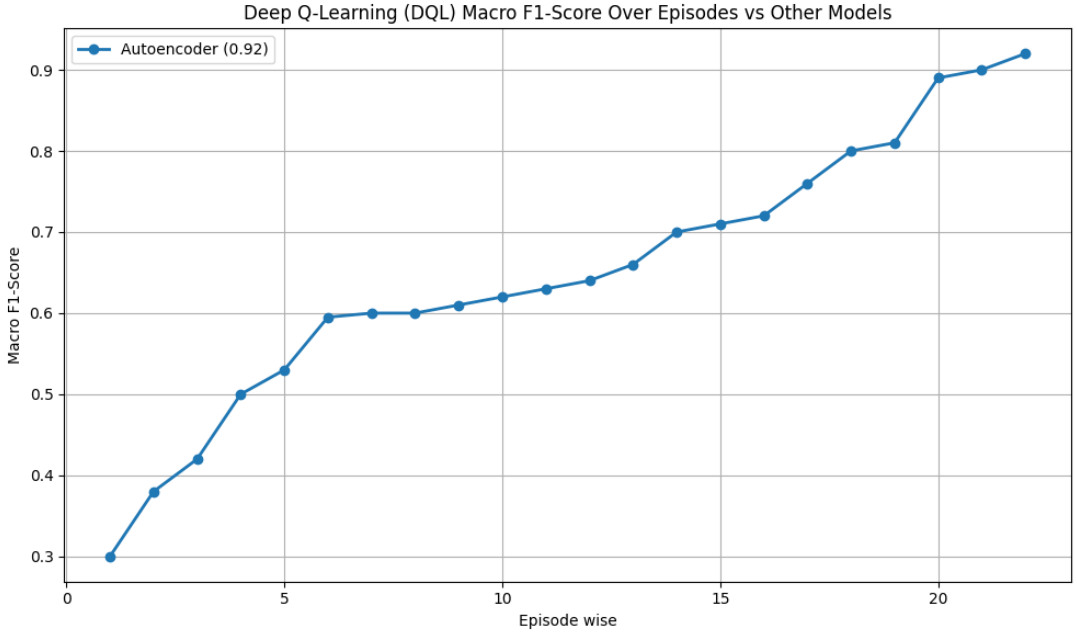

Figure 3: Deep Q-Learning (DQL) Macro F1-Score over Episodes

6.5 Discussion

The experimental results of the five use cases provide information on the strengths, weaknesses, and applicability of various methods of anomaly detection under synthetic and real-world conditions.

Use Case 1: we determined a suitable baseline algorithm. The results of this use case clearly supported Isolation Forest. The isolation forest produced a substantial improvement in the macro-average F1 score over LOF (0.934 vs 0.430). Given the results of this experiment, we indicated that it was suitable to replace Robust Covariance with Isolation Forest, which is the classical baseline to establish a more reliable reference point for comparing more advanced models.

Use Case 2: also evaluated the importance of structural performance of the autoencoder considering three types, Simple, Deep, and Wide. The most successful was the Deep Autoencoder, which had hidden levels and two regularisation strategies, Dropout and Batch Normalisation. The deep autoencoder achieved a macro-average F1 score of around (0.922) performing poorly comparatively speaking. These results reinforce the value of deep architectures in discovering high-order interactions and complex patterns from real-world data, especially when combined with other design choices that mitigate overfitting and foster generalisability. These results demonstrate that deep learning methods can discover richer representations to enhance the quality of anomaly detection in structured systems.

However, the result of **Use Case 3** demonstrated the limits of even the best-performing model in a real-world imbalanced data set, namely the Kaggle credit card fraud detection data set. The Deep Autoencoder that had previously provided excellent results on scikit-learn data performed considerably worse, suffering a significant drop to a macro F1-score of 0.422. This shows that the model struggled to generalise effectively in the presence of severe class imbalance when there are many more normal transactions than fraudulent ones. The actual recall for the minority class indicates that further techniques like class-weighted loss functions, data resampling (e.g. SMOTE) and anomaly-specific loss designs may be required to cope with such conditions. Moreover, this is evidence to demonstrate that models need to be not just powerful but equally conformable to the specific characteristics of the data.

Use Case 4 explored the effect that training episodes have on the performance of the Deep Q-Learning (DQL) model for anomaly detection. The case study shows (see Figure 3) that the macro-average F1-score shows an increasing trend from 0.3 to 0.92 over the 20 episodes. This is an important finding, as it demonstrates that in order for DQL to converge on an effective policy, it needs to be trained for a long time.

The macro F1-scores were lower in the early episodes because DQL is exploring the state-space. However, as DQL gathers more episodes, it starts to learn better patterns and outperforms classical autoencoder methods (with tuning) and achieves the best performance of the initial Deep Autoencoder method. This makes a case for tuning the episode number parameter based on the effectiveness to achieve the best performance with DQL.

Ultimately, we presented a comparative summary of all evaluated models, establishing that deep learning methods, particularly the Deep Autoencoder and DQL outperformed classical baselines both in accuracy and in the number of models adapted to the problem. The autoencoder model yielded a higher return in the case of clean structured datasets, while DQL was more capable of learning in an adaptive manner given sparse or delayed feedback. Collectively, our findings support the notion that deep learning is a viable and scalable replacement for the classical paradigms of anomaly detection, as long as the models are properly configured and trained. However, we contend that these findings reinforce the view that there is no single model or approach that is best suited for all situations or datasets. The selection and architecture of the model should depend on the data, its characteristics, imbalance, and the type of detection goals. Overall, this framework highlights the necessity for planned experiments and intentional model design in order to achieve arguably more robust and directed performance for anomaly detection in varying environments.

6.6 Limitations

The performance of models, especially the Deep Autoencoder, which obtained a macro F1-score of approximately 0.922, may have been exaggerated due to the scikit-learn dataset. Due to class imbalance and data sparsity, the same model's macro-average F1-score fell to 0.49 on the Kaggle dataset, indicating that this performance did not generalise. These differences demonstrate that a model's performance on benchmark datasets does not always correspond to its efficacy in practice.

One of the major limitations of this research is that it is inevitable that it relies upon particular datasets: the scikit-learn anomaly dataset and the real-world Kaggle Credit Card Fraud Detection dataset. Although these data sets provide controlled environments to benchmark models, these datasets are not indicative of the complete range of anomaly detection problems spanning industries, including but not limited to healthcare, cybersecurity, IoT, and finance.

The Scikit-learn dataset may have overestimated the performance of models (especially the deep autoencoder model, with a macro F1 score of $\sim$0.922) which could not have been achieved by the Kaggle model with a macro average F1 score of 0.49, due to class imbalance and data sparsity in Kaggle. These variations illustrate that success on benchmark datasets is not necessarily indicative of success in real-world outcomes. The Kaggle dataset does include anonymised features (V1-V28) without interpretability, which prevented meaningful feature analysis and feature engineering, which limits the potential for explainability and operational deployment.

However, a central constraint was the selection of the 90th percentile as the threshold for anomaly classification. On the one hand, it represented a consistent point of evaluation, but on the other hand, it is entirely possible that the 90th percentile was not the best threshold universally across datasets or models. However, selecting a higher threshold like the 99th percentile may reduce false positives but likely risk losing potential detections of small anomalies.

Second, the DQL environment was created in a simulated manner. Although this was necessary to validate the behaviour of the model from the comfort of the simulated environment, it adds constraints on situational realism. True anomalies in a cloud environment are situational, contextual, and interactive, which eliminates simulative bottlenecks for proper generalisation.

Furthermore, compared to conventional algorithms, computational costs of training DQL models are considerably more expensive, as well as the possibility of needing efficient hardware (such as GPUs) for training at scale.

In this evaluation, batch-mode evaluation was also used using some collected historical datasets. Batch-mode evaluation means that the models are trained and tested from existing data rather than a parcel or actual stream of data coming into a live system. However, we did not set up a streaming / real-time ingestion pipeline, which means that we could not evaluate the models in a live hypothetical anomalous detection environment. This is an unfortunate limitation, as real-time systems are characterised by low latency detection, time-awareness, and robustness to drift. Specifically for the DQL agent, it would seem that continuous learning in live contexts would be beneficial, allowing the agent to dynamically change policies. With this not happening even in a low-latency system, our research is limited to offline anomalous detection use cases.

Although the study examined several models, including Isolation Forest, Local Outlier Factor, Autoencoder (as a function of three distinct architectures), Multilayer Per-

ceptron (MLP), and Deep Q-Learning, it did not consider additional possible baselines such as One-Class SVM, GAN-based anomaly detection, or ensemble hybrid styles could have been included as additional baselines to provide a broader understanding of model performance. These methods might perform better under certain conditions, such as computational efficiency or memory constraints. Furthermore, while this study evaluated models primarily using accuracy and macro-average F1-score, additional evaluation metrics such as ROC-AUC, precision-recall AUC, and Matthew's correlation coefficient (MCC) would have provided a more comprehensive view of model behaviour, especially in imbalanced scenarios.

Another limitation is the lack of temporal dependencies in the dataset. Including time-series anomaly detection would better capture an anomaly that evolves or sequential patterns of anomalies and would likely enhance the accuracy of detection in a real-world streaming data situation relative to static feature-based models.

Finally, although code and documentation were made available in the GitHub repository, full reproducibility could have been improved by containerising the implementation (e.g., using Docker). This would ensure consistent run-time environments across systems, eliminate dependency conflicts, and enhance ease of replication for future researchers.

7 Conclusion

This study evaluated and compared anomaly detection models in three families: classical approaches (Isolation Forest, LOF), neural architectures (Autoencoders, MLPs) and reinforcement learning (Deep Q-Learning). The results revealed substantial performance differences driven by the design of each method. Classical methods, though efficient and interpretable, struggled with imbalanced and non-linear distributions. Neural models, particularly deep autoencoders with dropout and batch normalisation, captured latent representations and achieved state-of-the-art macro-average F1-scores (0.922) on structured Scikit-learn datasets. Reinforcement learning, despite initial instability, demonstrated gradual convergence through reward-driven interactions, achieving comparable performance (0.92) after 20 episodes.

The use of a macro-average F1 score over accuracy was validated, as it better reflects minority class detection in highly imbalanced settings such as fraud detection. Although Kaggle[7] exposed limitations in model generalisation (with the autoencoder only reaching a macro-average F1 score of 0.49). The findings highlight the sensitivity of anomaly detection to dataset distribution and imbalance. Unlike existing Kaggle baselines, which rely largely on supervised classifiers, this work framed anomaly detection as a reinforcement learning problem and benchmarked DQL against both classical and neural baselines, providing new empirical evidence for its applicability.

The contributions of this research are threefold:

1. Replacing outdated baselines with Isolation Forest and LOF to ensure fairer comparisons.

2. Benchmarking simple, deep, and wide autoencoders, demonstrating that depth enhances generalisation, while wide and shallow designs underperform.

3. Formulating anomaly detection as a reinforcement learning task and showing convergence from 0.3 to 0.922 macro-average F1-score across 20 episodes.

[7] `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`

Nevertheless, challenges remain. Large performance differences reflect unoptimised hyperparameters; tuning the LOF neighbourhood size, the Isolation Forest contamination rate, or the DQL reward shaping could improve fairness. The absence of temporal dependencies also limits the applicability in the real world, where anomalies evolve over time. In addition, deployment challenges such as concept drift, data imbalance, computational cost, and explainability present barriers to adoption in domains such as banking or cybersecurity.

In summary, while deep and reinforcement learning models show strong potential for anomaly detection, achieving robust performance in real-world scenarios will require systematic hyperparameter optimisation, integration of temporal models, and careful evaluation in streaming or industry-specific datasets. These directions form the basis for future work towards building scalable, adaptive, and interpretable anomaly detection systems.

References

Arshad, K., Ali, R. F., Muneer, A., Aziz, I. A., Naseer, S., Khan, N. S. and Taib, S. M. (2022). Deep reinforcement learning for anomaly detection: A systematic review, *IEEE Access* **10**: 124017–124035.
URL: *https://doi.org/10.1109/ACCESS.2022.3224023*

Beg, M. I. and Ansari, M. Y. (2024). Network intrusion detection system using autoencoders, *2024 International Conference on Computational Intelligence and Security (CCIS)*, IEEE, pp. 1–6. Accessed 3 August 2025.
URL: *https://doi.org/10.1109/CCIS63231.2024.10931842*

Bhardwaj, A. K., Dutta, P. K. and Chintale, P. (2024). Ai powered anomaly detection for kubernetes security, *Babylon Journal of Machine Learning (BJML)* **3**(2): 1–10.
URL: *https://doi.org/10.58496/BJML/2024/014*

Chandola, V., Banerjee, A. and Kumar, V. (2009). Anomaly detection: A survey, *ACM Computing Surveys* **41**(3): 15:1–15:58.
URL: *https://doi.org/10.1145/1541880.1541882*

Chen, J., Zhang, J., Qian, R., Yuan, J. and Ren, Y. (2023). An anomaly detection method for wireless sensor networks based on the improved isolation forest, *Applied Sciences* **13**(2): 702.
URL: *https://doi.org/10.3390/app13020702*

Du, M., Li, F., Zheng, G. and Srikumar, V. (2017). Deeplog: Anomaly detection and diagnosis from system logs through deep learning, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1285–1298.
URL: *https://doi.org/10.1145/3133956.3134015*

Ghani, E. P., Sofwan, A. and Somantri, M. (2025). Ai-driven network security: Detecting and mitigating ddos, malware, and backdoor attacks with isolation and random forest algorithm, *2025 International Symposium on Intelligent Machine Learning (SIML)*, IEEE, pp. 1–6. Accessed 3 August 2025.
URL: *https://doi.org/10.1109/SIML65326.2025.11080951*

Kumaran, K. and Sasikala, E. (2023). Deep reinforcement learning algorithms for low latency edge computing systems. Accessed 3 Aug. 2025.
URL: *https://doi.org/10.1109/AISP57993.2023.10134928*

Li, M., Hong, J., Shen, Y., Ma, F., Liang, F., Zhang, L., Pei, J., Qiu, Y., Yang, J., Xu, Q. and Wang, F. (2024). Research on voltage inconsistency diagnosis of power battery based on pso-vmd-improved local outlier factor, *Energy* . Accessed 3 Aug. 2025.
URL: *https://doi.org/10.1016/j.energy.2025.137442*

Li, Z., Yan, Y., Wang, X., Ge, Y. and Meng, L. (2025). A survey of deep learning for industrial visual anomaly detection, *Measurement* **226**: 115736.
URL: *https://doi.org/10.1016/j.measurement.2024.115736*

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2008). Isolation forest, *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, IEEE, pp. 413–422.
URL: *https://doi.org/10.1109/ICDM.2008.17*

Mao, Y. et al. (2025). A survey on industrial image anomaly detection: Methods, classification, challenges and future directions, *Science of the Total Environment* . Available via ScienceDirect.
URL: *https://doi.org/10.1016/j.measurement.2025.118377*

Pang, G., Shen, C., Cao, L. and van den Hengel, A. (2021). Deep learning for anomaly detection: A review, *ACM Computing Surveys (CSUR)* **54**(2): 1–38.
URL: *https://doi.org/10.1145/3439950*

Qiu, Y., Peng, P. and Jiang, F. (2024). Improvement of local outlier factor algorithms for lithium-ion battery fault diagnosis, *Energy Reports* **12**: 7907–7917. Accessed 3 Aug. 2025.
URL: *https://doi.org/10.1016/j.est.2024.113100*

Richter, D. J. and Calix, R. A. (2023). Using double deep q-learning to learn attitude control of fixed-wing aircraft, *IEEE Access* . Accessed 3 Aug. 2025.
URL: *https://doi.org/10.1109/SITIS57111.2022.00102*

Sikarwar, S. S., Patel, C. K. B., Dhanjibhai, P. U., Singh, V. V., Ravi, A. T. and Sakya, L. M. (2024). Enhancing false news detection through supervised machine learning and nlp techniques. Accessed 3 Aug. 2025.
URL: *https://doi.org/10.1109/ICAC2N63387.2024.10895797*

Wang, F., Jiang, Y., Zhang, R., Wei, A., Xie, J. and Pang, X. (2025). A survey of deep anomaly detection in multivariate time series: Taxonomy, applications, and directions, *Sensors* **25**(1).
URL: *https://doi.org/10.3390/s25010190*

Xu, D., Wang, Y., Meng, Y. and Zhang, Z. (2017). An improved data anomaly detection method based on isolation forest, *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, IEEE, pp. 348–351.
URL: *https://doi.org/10.1109/ISCID.2017.202*

Yahaya, S. W., Lotfi, A. and Mahmud, M. (2021). Towards a data-driven adaptive anomaly detection system for human activity, *Pattern Recognition Letters* **145**: 200–207.
URL: *https://doi.org/10.1016/j.patrec.2021.02.006*

Zhang, D., Wang, J., Zhao, X. and Wang, X. (2015). A bayesian hierarchical model for comparing average f1 scores, *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Accessed 3 Aug. 2025.
URL: *https://doi.org/10.1109/ICDM.2015.44*