CMPT 318 Term Project

Group 13 Fall 2023

Anmol Sangha 301394707

Nav Gosal 301360678

Vinesh Reddy 301357491

Jonny Lin 301380640

Abstract

This project delves into the realm of cybersecurity challenges faced by supervisory control systems, essential for the smooth functioning of modern infrastructure. Focusing on unsupervised intrusion detection, the project employs Hidden Markov Models (HMMs) and Principal Component Analysis (PCA) on a dataset of electrical consumption. By emphasising feature selection, model optimization, and precise anomaly identification, the objective is to fortify supervisory control systems against cyber threats. Striking a balance between precision and recall in anomaly detection becomes pivotal for practical application in real-world scenarios. Through the integration of HMMs and PCA on an electrical consumption dataset, this project endeavours to contribute to the robust defence of critical infrastructure in an era of heightened digital connectivity.

Table of Contents

Table of Figures

1 Problem scope

Supervisory control systems play an indispensable role in the seamless functioning of our modern world. They're the invisible hands that keep our lights on, water clean, and trains running on time. These systems manage and oversee automated processes across multiple industries, ensuring that everything from manufacturing plants to utility services operates efficiently and smoothly. However, the critical nature of these systems makes them a key target for cyber attacks. Imagine the mayhem that would occur if a malicious attacker gained control over a city's power grid or a country's rail network or water treatment facilities. Such scenarios are real threats that organisations and governments face every day. An attack on these systems can have wide-reaching consequences, including service disruptions, economic impacts, and even risks to public safety. The ripple effects of such incidents can be far-reaching, impacting even national security. Recognizing the gravity of these threats, it's crucial for organisations to fortify their supervisory control systems against cyber attacks. This isn't a one-time fix but a continuous process of vigilance and improvement.

Although it is impossible to be 100 percent safe from a cyber attack, some ways to protect from cyber attack are network segmentation, regular updates and advanced threat detection systems. Network Segmentation involves dividing the control system network into separate, secure zones. This strategy limits the spread of an attack, ensuring that a breach in one area doesn't compromise the entire system. Regular updates and patch management are also vital. Just as we update our smartphones and computers to protect against the latest threats, the same principle applies to these control systems. By keeping software up-to-date, organisations can guard against known vulnerabilities that hackers might exploit. Advanced threat detection systems are another key component. These systems are like high-tech watchguards, constantly scanning for unusual activity that might indicate a cyber intrusion. If something suspicious is detected, these systems can alert human operators, who can then take swift action to mitigate the threat. supervisory control systems are the backbone of our critical infrastructure, but their importance also makes them a target. By investing in robust cybersecurity measures, organisations can protect these systems, ensuring that our modern way of life continues uninterrupted. It's a challenging task, but in an increasingly connected and digital world, it's absolutely essential.

2 Problem being addressed

This project involves addressing critical challenges in cybersecurity, specifically focusing on unsupervised intrusion detection in supervisory control systems. Since these systems are pivotal in the functioning of critical infrastructure, their security is paramount. This project addresses key problems in developing an advanced anomaly detection system for supervisory control systems, focusing on feature selection, model optimization, and effective anomaly identification. Solving these problems helps us develop robust anomaly detection models that can identify potential intrusions or irregularities in these systems, and thereby enhancing their resilience against cyber attacks. A crucial aspect of the project is finding a balance between precision and recall in anomaly detection. Reducing the false alarm rate is essential to make anomaly detection practical and effective in real-world scenarios, especially under resource constraints.The project’s approach needs to be robust enough to handle these complexities while remaining efficient and accurate.

3 Methodology

Addressing the critical need for enhanced cybersecurity in supervisory control systems, our methodology involves developing a sophisticated anomaly detection system that emphasises feature selection, model optimization, and precise anomaly identification. Initially, we will employ Principal Component Analysis (PCA) to sift through extensive datasets, pinpointing the most relevant features. This step is crucial for streamlining the model, thus reducing computational load while boosting accuracy in detecting anomalies. Subsequently, we focus on constructing and fine-tuning Hidden Markov Models (HMMs) using these selected features. The effectiveness of these models is carefully assessed through log-likelihood measurements and the Bayesian Information Criterion (BIC), guiding us in choosing the most proficient model. This carefully calibrated model is very skillful at discerning irregularities with great precision. Strict testing will further refinement of our system against datasets containing simulated anomalies. In this process, calculating the log-likelihood for different observation sequences is key to distinguishing real anomalies from normal variations. This approach significantly reduces false positives, making our anomaly detection methodology not only more practical but also highly effective in real-world settings. Ultimately, our thorough methods come together to create a strong and efficient system for detecting anomalies. This system effectively identifies potential security issues and strengthens the defence of supervisory control systems against cyber threats. It also achieves a good balance between accuracy and the ability to identify true threats across various complex situations.

4 Characteristics and rationale

The first section of the script sets up the environment, imports data, performs initial data quality checks, handles missing values, and begins data preprocessing by standardising values. This is also where our first design choice was made. To handle the missing values we chose to eliminate them instead of interpolate them. Several factors influenced this decision, first of which was to maintain the integrity of the dataset. Estimating missing values through interpolation requires drawing conclusions from existing data, a process that might not always reflect the actual or representative missing values accurately. Secondly, Interpolation can introduce bias, especially if the missing data is not random. If there’s a systematic reason behind the missing values, interpolating them could skew the results and lead to incorrect conclusions. Lastly, Interpolating values, depending on the method used, can be computationally intensive. Elimination of N/A values reduces this burden, which can be significant in large datasets like the one we are using.

4.1 Rationale for splitting train data and test data

In our approach, we strategically divided the dataset into two distinct segments, allocating 80% for training purposes and reserving 20% for testing. This decision was made by looking at several key considerations, aimed at optimising the performance and accuracy of our model.

First and foremost, dedicating 80% of the data to training is important in ensuring comprehensive model training. This substantial proportion is essential for the model to effectively comprehend and understand the complex patterns and connections within the data. It’s akin to providing a robust and extensive base for the model, allowing it to build a deep and nuanced understanding of the dataset.

Moreover, this 80/20 division serves as a critical balance point in overcoming two common pitfalls in data modelling: bias and variance. Bias, often a consequence of oversimplified models, that can lead to a fundamental misrepresentation of underlying trends. On the other hand, variance, typically arising from overly complex models, can cause the model to be excessively influenced by minor changes in the training data. The 80% training portion is substantial enough to significantly diminish the risk of bias by providing a varied dataset for the model to learn from. Simultaneously, the 20% testing data is appropriate to effectively gauge the model’s performance on new, unseen data, thus helping in controlling variance.

Furthermore, this split is a pragmatic choice considering computational efficiency. Training models, especially complex ones, can be a resource-intensive process. By limiting the training data to 80%, we strike a sensible balance between ensuring sufficient training depth and maintaining computational feasibility.

In summary, our chosen data partitioning scheme of 80/20 is a well-considered strategy that not only fosters robust and well-rounded training data for the model but also ensures efficient and effective model evaluation, while sensible managing computational resources. This approach, therefore, enhances the overall integrity and reliability of our modelling process.

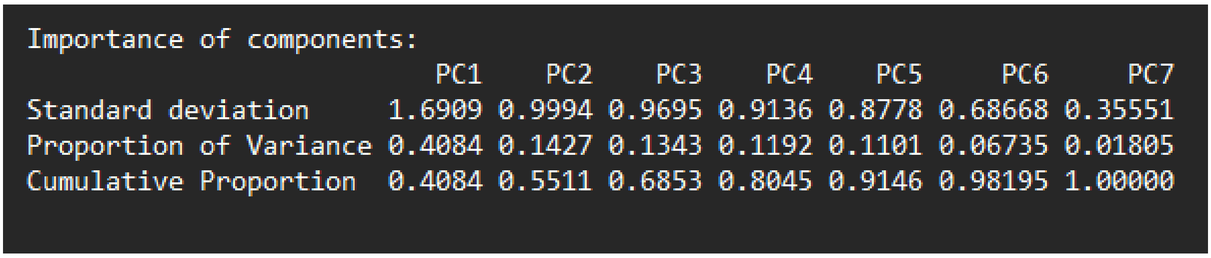
principal components

Upon completing the Principal Component Analysis (PCA), we gain access to vital metrics that guide the selection of response variables from the principal components. Figure 1.1 illustrates the PCA summary. It starts with the standard deviation for each principal component, a measure showing the extent of variability each component encompasses. The greater the standard deviation, the more variability the component is capturing. In our dataset, PC1 emerges as the dominant component with the highest standard deviation at 1.6909, indicating its significant role in explaining the dataset's variability.

Next, the summary presents the proportion of variance, which quantifies the share of the dataset's total variance that each component explains. Here, PC1 stands out by accounting for 40.84% of the variance, noticeably more than its counterparts. The final key metric is the cumulative proportion, representing the aggregated percentage of variance that the principal components explain cumulatively. This metric is helpful in determining the number of components to retain for optimal data representation. For instance, the combination of PC1, PC2, and PC3 explains 68.53% of the total variance.

Additionally, for greater clarity, a bar graph (refer to figure 1.2) visually depicts cumulative proportion, offering a more intuitive understanding of each component's contribution. Based on these insights, PC1 clearly takes priority over as the most influential component in terms of variance captured. The subsequent components, such as PC2 and PC3, contribute to a diminishing share of the variance. In our PCA approach, we focus on the cumulative proportion to maintain a balance between capturing a substantial portion of the total variance and minimising the number of variables. Selecting PC1 to PC3 encompasses 68.53% of the variance, generally sufficient for a broad range of analyses but extending this selection to include PC4 elevates the covered variance to 80.45%, thereby achieving a more thorough representation of the data at the expense of incorporating an additional variable.

Figure 1.1.1



```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7
Standard deviation     1.6909 0.9994 0.9695 0.9136 0.8778 0.68668 0.35551
Proportion of Variance 0.4084 0.1427 0.1343 0.1192 0.1101 0.06735 0.01805
Cumulative Proportion  0.4084 0.5511 0.6853 0.8045 0.9146 0.98195 1.00000
```

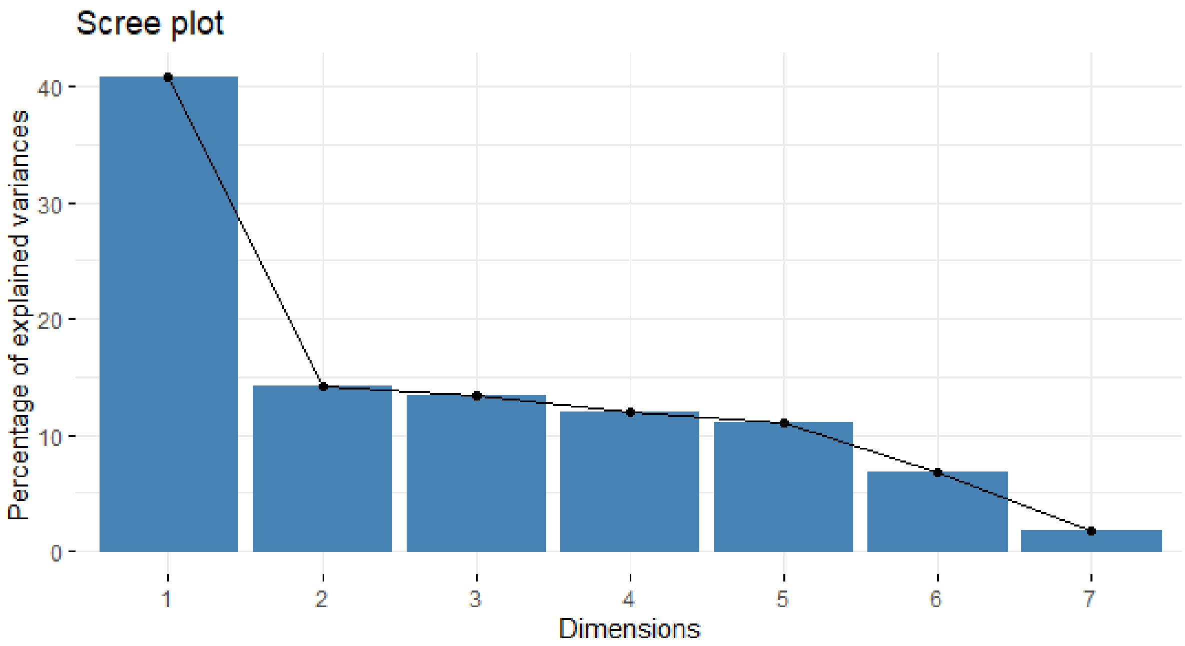


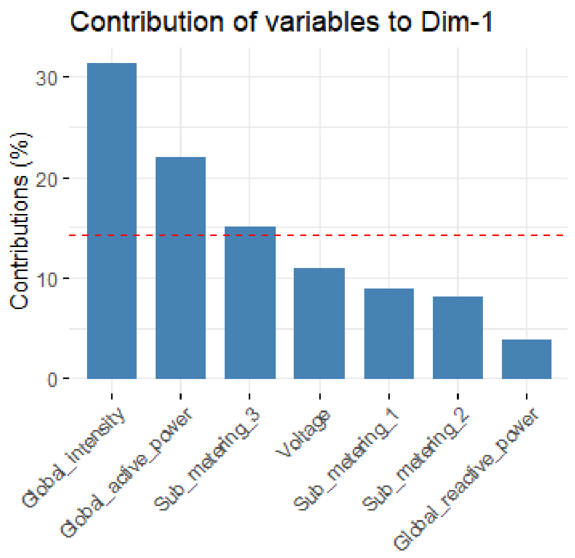
Figure 1.1.2

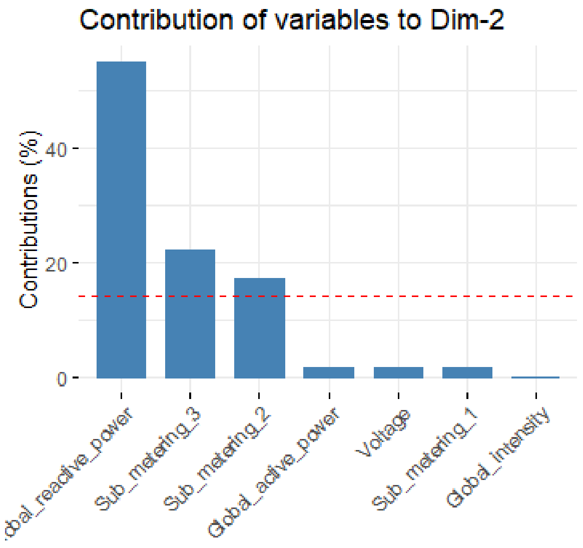
4.2 Comparing response variables

In Principal Component Analysis (PCA), an important step is comparing each response variable with others to see how much they contribute to the variance shown by the principal components. This examination is critical in determining the variables to be retained for subsequent analysis or modelling purposes. A paramount aspect of this evaluation lies in understanding the loading scores, also known as rotation data. These loadings serve as indicators of the extent to which each variable influences a particular principal component. Notably, a high loading, approaching 1 or -1, indicates a great impact of the variable on the component.

Looking further into loading scores, we see the variables having significant sway over the dataset's variance. For instance, as shown in figure 1.2.1, the loading source data are scaled to represent the proportionate contribution of each response variable to each principal component. Each graph represents a principal component. In the case of PC1, labelled 'Contribution of variables to DIM-1,' it's observed that 'global intensity' accounts for over 30% of this component, while 'global active power' contributes to just over 20%.

Using scaled rotation data from graphs helps with more than just seeing the data.; it enables a systematic ranking of the response variables in terms of their impact. Following this method, 'Global_active_power,' 'Global_intensity,' and 'sub metering 3' emerge as the most impactful. This kind of analytical approach not only yields a quantitative understanding of each variable's influence but also helps in making informed decisions about which variables are most relevant for the focus of analysis.





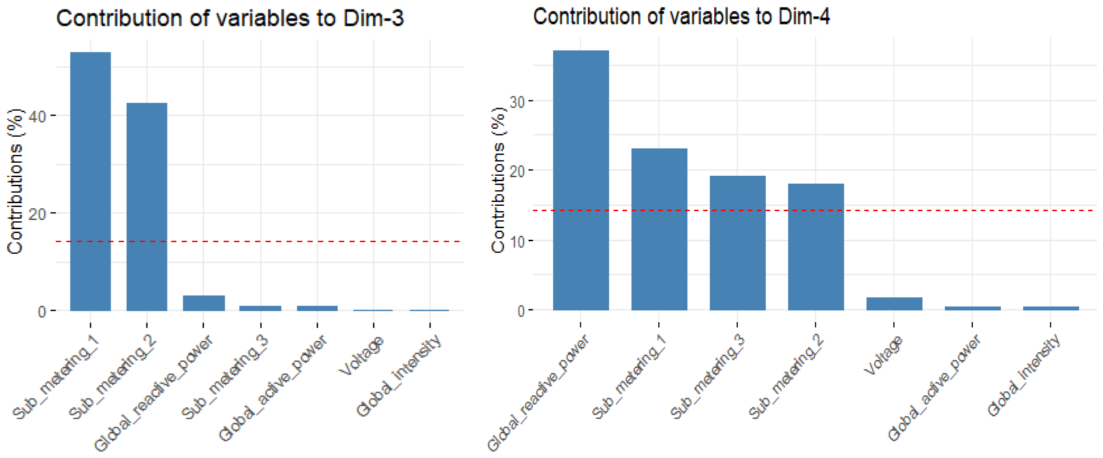


Figure 1.2.1

4.3 Selection of response variable and time window

The variable 'Global_active_power' with a substantial loading of 0.4684520 on the principal component PC1. This component alone accounts for 40.84% of the dataset's variance, highlighting the variable's crucial role. Similarly, 'Global_intensity' also exhibits a notable loading on PC1 (0.5597089), further emphasising its importance in explaining the dataset's variance. Additionally, 'Sub_metering_3' emerges as a key variable with the highest loading on PC2 (0.47145946). Given that PC2 represents an additional 14.25% of the variance, this inclusion adds a valuable dimension in understanding different facets of the data.

The rationale behind selecting these variables is twofold. Firstly, their significant loadings on the principal components indicate their strong influence in the dataset, capturing diverse aspects of electricity usage. This diversity is crucial for constructing a robust Hidden Markov Model (HMM). Secondly, by prioritising variables with high loadings on the first two principal components, the model is better positioned to encompass a significant portion of the dataset's variability. This aspect is pivotal for effective anomaly detection within the context of electricity consumption.

In conclusion, the chosen variables ('Global_active_power', 'Global_intensity', and 'Sub_metering_3') offer a detailed perspective on the variance within the dataset. Their selection, guided by PCA results, ensures a data-driven foundation for developing an HMM. This model is thus well-equipped to identify anomalies in electricity consumption, leveraging statistical insights for enhanced accuracy.

4.4 Final model selection

An integral aspect of model evaluation, log-likelihood measures how well the model explains the observed data, in comparison to our other models. Higher values indicate a better fit of the model to this said data. In addition, we also utilized the Bayesian Information Criterion (BIC). This metric evaluates the balance between the fit and the complexity of the model. It includes a penalty term for model complexity, which assists us in identifying the most suitable model. Our process for selecting models takes into account both log-likelihood and BIC values to obtain a balanced selection where we can capture the underlying patterns in the data without overcomplexity.

In our model selection process, we conducted tests on Hidden Markov Models (HMMs) with various numbers of states (nstates). Early experiments with a range of states (4-7) revealed substantially lower log-likelihood (logLike) scores, and high BIC scores, indicating poor model fit. We found well-performing models in the 8-10 range, however we noticed a slight dip in logLike from state 9 to state 10, as well as an increase in BC. This information is displayed below with log-likelihood in green and BIC in red.

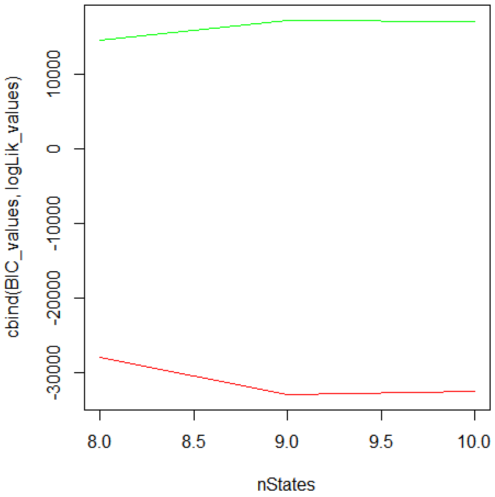


Figure 2.1.1

We continued on, however we then encountered convergence issues for models with nstates greater than 10. These indicators displayed signs of overfitting. The above information led to the selection of models with states 8,9 and 10 to use on the test data in the following calculations.

4.5 Comparing log-likelihood

In our comparison of model performance, we calculated the normalized training log-likelihood and test log-likelihood across our distinct model configurations, testing with nstates equal to 8, 9, and 10. Analyzing the results, we observed that the model with nstates = 9 consistently yielded the highest normalized log-likelihood values for both the training and test datasets. This suggests that the model with nine hidden states aligns most closely with the underlying patterns in the data, demonstrating robust performance across both training and unseen test instances. Therefore, based on the superior performance indicated by normalized log-likelihood, we have selected the model with nstates = 9 as our final model.

4.6 Illustrating anomalies

The assessment of log-likelihood scores across the three datasets containing injected anomalies underscores the striking abnormality present within each. In scrutinizing the results, it becomes evident that all three anomaly datasets exhibit remarkably low log-likelihood scores, indicative of a substantial departure from the expected patterns modelled by our HMM. Remarkably, the log-likelihood score for the anomaly3 dataset not only registers as quite low, but even goes to NaN, highlighting an extreme and highly anomalous behavior. Meanwhile, in comparing results from the anomaly1 and anomaly2 datasets, anomaly2 displays comparatively higher log-likelihood values, suggesting a slightly less pronounced deviation. This gives us some direction that the anomaly1 dataset has a higher degree of anomalies present. In all, the overall trend across all datasets unmistakably signifies a substantial departure from normal patterns. These persistently low log-likelihood scores collectively underscore the presence of anomalies within the given observation sequences

5 Conclusion

In conclusion, our comprehensive investigation into enhancing cybersecurity for supervisory control systems through advanced anomaly detection has yielded significant findings. We finish this project with a clear understanding of the critical role these systems play in maintaining our modern infrastructure and the severe implications of their compromise through cyber attacks. Throughout this project, a key lesson we learned was the importance of balance in cybersecurity solutions. We discovered that it's crucial to accurately detect real threats while minimising false alarms. This balance is essential for maintaining trust in the system and ensuring that security measures are both effective and efficient. Another important takeaway was the value of continuous adaptation and learning in the face of evolving cyber threats, highlighting the need for systems that can adapt and improve over time. Our approach, based on data science and cybersecurity, involved several key steps: choosing important features through Principal Component Analysis, improving our model with Hidden Markov Models, and thorough testing against fake anomalies. By successfully identifying and implementing key variables like 'Global_active_power', 'Global_intensity', and 'Sub_metering_3', and carefully tuning our model to balance precision and recall, we've developed an efficient and effective anomaly detection system. This system effectively spots real threats while avoiding false alarms, improving the reliability of control systems.

List of References

StatQuest with Josh Starmer. (2017, December 4). StatQuest: PCA main ideas in only 5 minutes!!! [Video]. Youtube. https://www.youtube.com/watch?v=HMOI_lkzW08

StatQuest with Josh Starmer. (2018, April 2). StatQuest: Principal Component Analysis (PCA), Step-by-Step [Video]. Youtube. https://www.youtube.com/watch?v=FgakZw6K1QQ

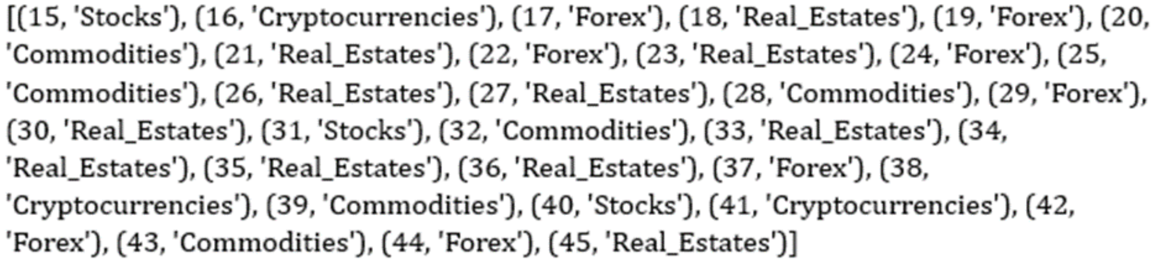
StatQuest with Josh Starmer. (2017, November 27) StatQuest: PCA in R [Video]. Youtube. https://www.youtube.com/watch?v=0Jp4gsfOLMs

Term Project Part 3

The training process uses three hyperparameters. Epsilon is used as a threshold in choosing whether to explore or use the trained data. This was set with a low value of 0.1 since exploring would have more random results due to the nature of profit calculations. The model would rely more on the training data to get an understanding of what is the better action to take at a given state. The alpha value affects the learning rate of the model. Considering the volatility of the market for investments, a relatively high alpha value of 0.8 was chosen so that the model does not get outdated in decision making. Since there is no concrete end goal for the investments other than to continuously gain rewards, the gamma value was set at 0.7. A long term reward is prioritised as that will ensure the company will continue to grow rather than being greedy in the short term. With investments however, sometimes a short term gain is necessary hence the gamma value was not set at 1. These decisions for the model should help employees make up-to-date decisions that will benefit the company in the long run.

It was found that setting the episode number at 10,000 was very time consuming so 2,000 was decided on. Note that the code still will take about 13 minutes to finish training. Overall the Q-Table doesn't seem to have a distinct pattern between the different actions across. In certain states, for example state 89 and 92 the difference in Q-values heavily favour one action over the others. Majority of the other states however have very close Q-values between actions. The Q Table generated can be shown in the provided QTable.csv file. Each row number is its state-1 as the first row was used to label the columns. The policy computed is shown in the Policy.csv file. Each row is associated with the corresponding state. The policy also reflects that overall there is no one area of investment that is dominantly profitable.

The MarketAgent function is used to test the unseen data. In the function, the optimal action table is created at each step of evaluating the unseen data. Below is the produced optimal action table stored in pairs of (state, action).



```
[(15, 'Stocks'), (16, 'Cryptocurrencies'), (17, 'Forex'), (18, 'Real_Estates'), (19, 'Forex'), (20,
'Commodities'), (21, 'Real_Estates'), (22, 'Forex'), (23, 'Real_Estates'), (24, 'Forex'), (25,
'Commodities'), (26, 'Real_Estates'), (27, 'Real_Estates'), (28, 'Commodities'), (29, 'Forex'),
(30, 'Real_Estates'), (31, 'Stocks'), (32, 'Commodities'), (33, 'Real_Estates'), (34,
'Real_Estates'), (35, 'Real_Estates'), (36, 'Real_Estates'), (37, 'Forex'), (38,
'Cryptocurrencies'), (39, 'Commodities'), (40, 'Stocks'), (41, 'Cryptocurrencies'), (42,
'Forex'), (43, 'Commodities'), (44, 'Forex'), (45, 'Real_Estates')]
```