**Title**

Isak Nyberg 1, Rahul Kothuri2

1Email Address (style ‘Affiliation) [isaknyb@kth.se](mailto:isaknyb@kth.se)

2Email Address (style ‘Affiliation) [kothuri@kth.se](mailto:kothuri@kth.se)

This material shall provide a description of the planned project. Present the main project in 5-6 pages A4-sized document.

It shall include following sections:

* Abstract
* Introduction:
  1. Background and Related Work (state of the art)
  2. Problem to be solved and Aims
* Contribution (Body text divided by numbered sub-headings):

Design

Architecture

Results

Evaluation

* Conclusions, Discussion
* References

Use headings to break the text.

Never use subtitles after each other without text in between the sections.

See more directions in Appendix A.

**Abstract**

*An abstract of 100-200 words in 10 point Arial Italic should be provided. The abstract is a concise summary of the project and the project results. The abstract also allows the paper to be catalogued, categorized and searched by subject and keywords (style ‘Abstract Italic’).*

Introduce the subject area for the project and describe the problem that is solved and described in this report. Present how the problems have been solved, and present results for the project.

The presentation of the results should be the main part of the abstract. Use about ½ A4-page.

Table of Contents

1 Introduction 1

1.1 Background and Related work (State-of-the-art and motivation) 1

1.2 Problems and aim 1

2 Contributions 2

3 Implementation 2

3.1 Design 2

3.2 Architecture 4

3.3 Results and Evaluation 4

4 Conclusions 4

5 References 4

Appendix A 1

# Introduction

Dota 2 is a so called Multiplayer online battle arena game (MOBA) developed by Valve Software. This type of videogame has two teams of 5 play against each other with the use of characters with certain abilities, in the case of Dota 2, the characters that each player uses is called a hero. The ultimate goal of the game is for one team to collect enough resources to be able to attack and destroy the enemy base while also defending their own base from the attacks of the enemy. While the game is mostly focused on the player versus player combat another very decisive part of the game is the heroes that are picked by each team. Due to the different abilities that the different heroes possess, their likelihood for either team to win can drastically differ depending on which heroes are picked by either team, this is known as the draft. This concept is what will be explored in this project.

## Background and Related work (State-of-the-art and motivation)

Dota 2 has already been explored with the use of artificial intelligence (AI). The company OpenAI cofounded among others, Elon Musk and Sam Altman in 2015 has already applied machine and deep learning to the game of Dota 2. Their project began in 2017 at the Dota 2 tournament The International 7 which is an annual Dota 2 tournament with multimillion-dollar price pools. At this event the bot made by the OpenAI team was able to beat one of the best human players in a one versus one match. This was the first time this had occurred.

<https://openai.com/blog/dota-2/> Two year later in 2019 OpenAI showed up at The International 9 where they managed to beat the first-place team in a 5 vs 5 match. Solidifying the claim that AI was superior to humans. <https://openai.com/blog/openai-five/>

While OpenAI primarily focused on the gameplay, having drafts either being done by humans or predetermined. For this project an understanding of how the drafting in Dota 2 is required. The game has a few different picking modes but the primary mode that is the one used in ~~Ranked All-Pick. The reason this will be used is because it is the most common type match played and this the easiest to sample data from. The match begins with both teams being presented with a screen of all 120 different heroes, from which each player gets to choose one hero to ban from the game, this means that the banned hero cannot be picked this match. Each ban has a randomized 50% change of going through and the same hero cannot be banned by two players. Thereafter the picking phase begins where any two players from each team can pick a hero, neither team can see what the opposing team has picked until the end of the phase. If the same hero happens to be picked by a player from both teams, there hero is banned instead and the phase restarts again. Thereafter another identical phase starts and two players from the remaining three on each team pick a hero. Thereafter the last phase occurs with the last player on each team picks. It is therefore strategic to save the most important hero for last since that way the opponent does not see what hero was picked and therefore cannot pick use that information when the pick a hero themselves.~~

## Problems and aim

The problem raised by the drafting is; how do does one determine what hero is the best to pick? Obviously the more informed the player is of the draft of the enemy but also the draft of the allied players the better decision the player can make to better counter the enemy and also suit their team. The way the player makes this decision is by so called counter picking his enemy. An example would be the hero known as Bounty Hunter, he wins on average 50% of the games that he is picked, but when he is played against another hero called Naga Siren, he only wins 46% of the time despite Naga Siren on average winning less than 50%. This means that Naga Siren is a counter to Bounty Hunter because when she is played on the opposite team Bounty Hunter has a lower chance of winning than what he usually has.

<https://www.dotabuff.com/heroes/bounty-hunter> most heroes have proximately 5% difference in win rate when played against their strongest counter however some heroes are very susceptible to being counter picked and have their win rate changed by more than 10%. In the given example it is quite trivial to compare the win probability of two heroes and determine which one is the better fit. However, the complexity increases significantly when including all of the eight heroes of the game will be sufficient as having to account for 5 opponents is more difficult, additionally it is also important that the hero pick also works well together with the rest of the team. Given that there is a hero pool of 120 different heroes the number of possible team combinations is . This number is far too big to brute force every combination. However, this raises the question *to what extent can deep learning be used to draft accurate heroes in Dota 2?*

To explore this question this project aims to use deep leaning to create a model that can be used to predict the best hero to pick given a draft from a Dota 2 game. This should be done with only information that a player can expected to know at the final pick of the draft (i.e. when 4 heroes have been picked by teammates and 5 have been picked by the enemy).

# Contributions

The demand for this type of product exists in both professional and amateur play. In Dota 2 tournaments a single coach is allowed to help the team during the draft. while the coaches are not allowed any digital help during the draft, this product can server as very good preparation. Additionally, amateur player who just want to increase their winning chances can use this product in order to gain an advantage. There already exist other applications that aim to give advantages in the draft. The betting scene for professional matches is also very big for Dota 2 and an individual could this product to evaluate the draft of either teams in order to place bets on which teams would win to increases the chance of winning the bet. It does however stand that this project does not have much use outside the Dota 2 community, however, is it possible to branch out the concept to other games within the same genre for example *League of Legends, Heroes of the Storm, Smite,* andotherMOBAgames.

It is to note that the game Dota 2 is dynamically updated every few months where heroes are changed, and new heroes are added. This means that, for the project to keep its relevancy new matches need to be collected and the network needs to be retrained otherwise some relationships between heroes may not hold true any longer.

This project does raise an ethical concern that it may be viewed as unfair to other players to use an AI to assist with the drafting. In order to ensure that this project does not the project in its trained from will not be publicly released.

# Implementation

## Requirements

Initially these are the requirements of the project:

* The project needs to be able to fetch data samples from Dota 2 matches.
* The fetched data needs to include matches from various skill groups
* The project needs to be able to discard samples that are not useful.
* The project needs to be able to reformat the samples and make it applicable for machine learning.
* The project needs to use machine learning to evaluate the samples based on the draft.
* The evaluation needs to have an accuracy that is equal or better than an experienced human.
* The project needs to be able to input a sample manually to be able to evaluate matches that are hypothetical or yet to be played.

## Design

The design of the project will be divided into smaller parts to ensure that the project does not become too complex. Each part deal with a specific set of requirements.

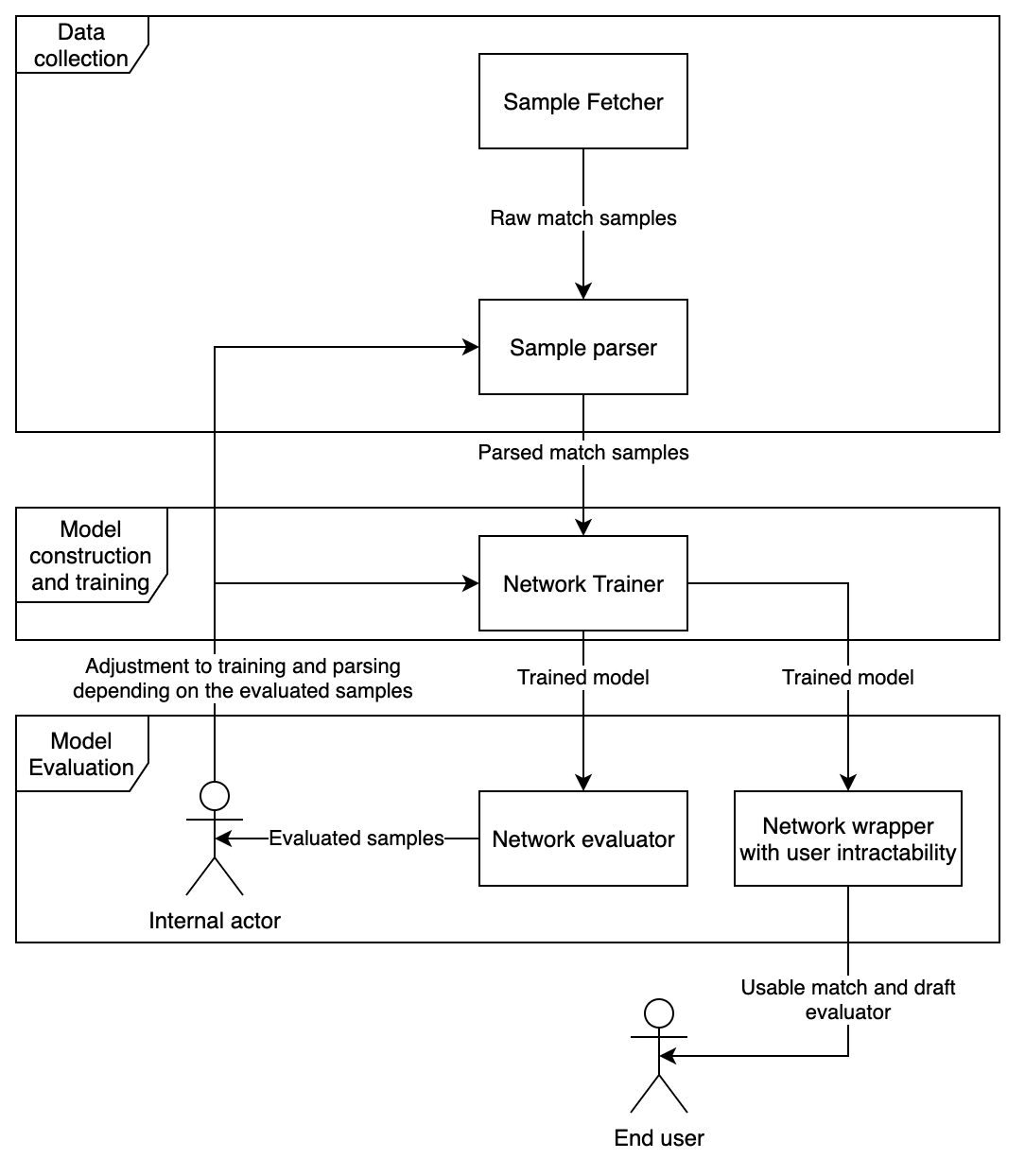


Figure 1 Project Design

Figure 1 illustrates the way that these requirements will be implemented. The project will be split into three main parts that all have different purposes in the project as a whole and the able to fulfil each requirement.

1. **Data collection**

Collected matches that will be used for the training of the network. The important parts of this is that the matches need to be diverse in their skill group, meaning that the model training should be based on matches played by both high- and normal skill players. They also need to be accurate matches played by real players, meaning that matches that are intentionally lost, contain bots or players who prematurely leave the game must be excluded. Like any deep learning project, a very large number of samples is needed.

1. **Model construction and training**

The model construction will consist of creating the nodes of the network and to reformat the data in a way that the model can efficiently interoperate. This part will also consist of the training of the model using the samples fetched in the data collection phase. This trained network will then be passed onto the evaluation. TODO ADD MORE HERE

1. **Model evaluation**

This phase will consist of the evaluation of the accuracy of the model. It will examine how well the model can evaluate a sample of matches compared to experienced humans. It will also asses what parts of the data that the network values as reliable indicators and how that creates to real matches. The evaluation will then be used to make improvements to the way that the data was parsed by for instance removing faulty samples, add more samples to reduce selection bias, or to increase the overall sample size. There may also be some adjustments to necessary to make to the network more accurate, such as changing the number of nodes or the number of hidden layers.

## Architecture

1. **Data collection**

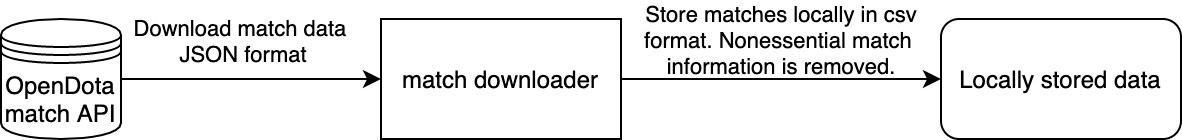
**

Figure 2 Data collection Architecture

The data collection is explained in figure 2. The project uses the OpenDota api (not to be confused with OpenAI) to download matches. This is a third-party service and not the official api provided by Valve Software. The reason this api was chosen is because OpenDota provides a greater access of past matches enabling the download of a greater number of matches. This has allowed for the download of just under 2 million matches for this project. That is approximately one week worth of matches. [https://www.dotabuff.com/heroes/played?date=week] OpenDota also allows for the download of match summaries, this means that just the essential information is contained such as the draft, match duration and rank of the players involved while leaving out information that is too match specific and not relevant to the project. This meant that the download of matches became a lot faster as OpenDota was tested to download approximately 100 matches per second compared to official Dota 2 Api provided by Valve Software which only allowed 10 matches per second. All of this led to the decision to use the OpenDota API for the data collection.

The locally stored data was formatted in a comma separate value format. The first value was the match-id which is a unique identifier for each match which can be used for debugging and in the evaluation of the network. The next 10 values are the heroes that played in the match sorted by team. Internally heroes also have a unique id that was used rather than storing the full name. The next value stored is the duration of the match in seconds, this value is useful since the duration of the match has a significant impact on what heroes that preform the best. The second the last value is the rank of the match, an integer between 15 and 80 with 80 being the more skilled players. The rank is important since some heroes require a lot more skill to play well and would thus perform better at higher ranks. An example of this is the hero Lycan who has a win probability of above 60% in the high skilled games while only having a win probability of 45% in low skilled games. [https://www.opendota.com/heroes/public] Lastly the label is stored to determine which team that won the game.

1. **Model construction and training**

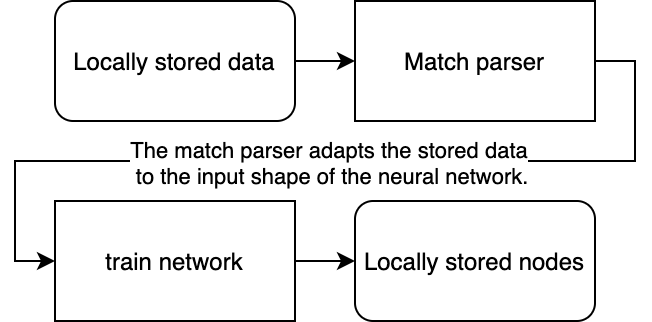
****

Figure 3 Data parsing Architecture

In figure 3 the stored data is reformed to fit the input shape of the neural network. The neural network had an input shape of 259 nodes. This shape was chosen because every hero needs two nodes to indicate its presence in the game one for each team, and additionally the match duration and rank. However, with only 120 heroes this adds up to 242 nodes, the difference is the nodes is due to how Dota 2 indexes hero-ids. There are some hero-id that for undisclosed reasons are not used, for example hero-id 24 meaning that the highest hero id is 129. As a result, there are some nodes in the network that are not used but the convince of being able to index the nodes by hero-id is outweighs the inefficiency of a few more unused nodes added.

The network only has one output layer which is the evaluation of the match. This value should tend towards 0 if the primary team is expected to lose the match and towards 1 if the primary teams is expected to win the match. This means that each downloaded match can be used twice, once from the perspective of the winning team and once from the perspective of the losing team. It also gives more nuance to the evaluation of a match as a value close to 1 or 0 is a very confident evaluation but a value that is closer to 0.5 (the expected value) is a less confident evaluation.

The network uses three hidden layers with 172 nodes each. The reason three hidden layers is simply because when evaluating it was the number of layers that yielded the best results of 62% accuracy. Additional layers may be more accurate if a larger data sample could be provided. However, the returns gained by added more layers are expected to be demising while training time grows exponentially. Additionally, an accuracy of 62% does lie within what is the personal estimated percentage of games that are determined by the draft while the remaining games are determined by the gameplay of the players. The reason each hidden layer has 172 nodes is because it is two thirds of the number of input nodes and since the is only one output node, it should be sufficient.

[<https://www.allaboutcircuits.com/technical-articles/how-many-hidden-layers-and-hidden-nodes-does-a-neural-network-need/>]

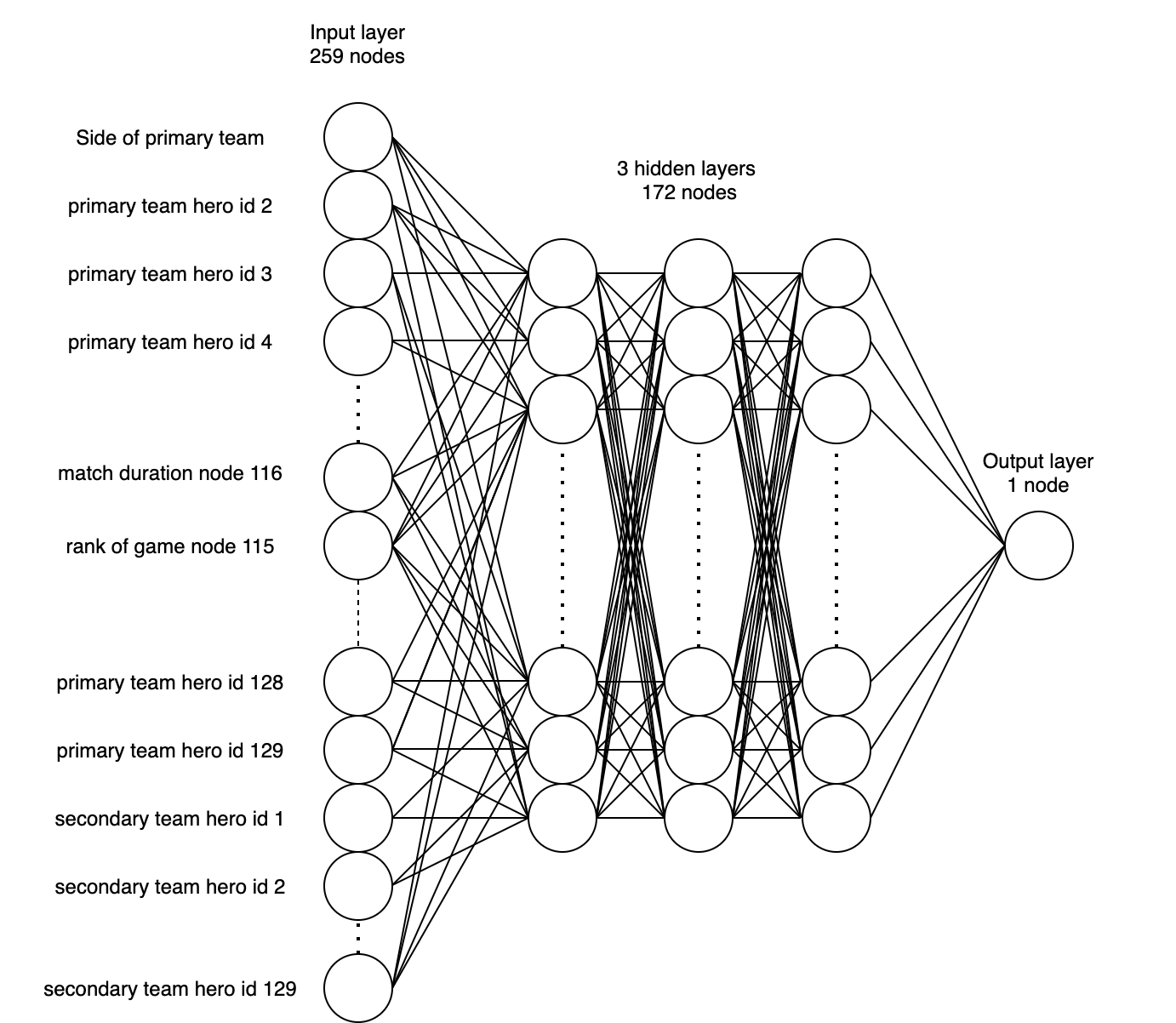


Figure 4 Neural Network Architecture

Figure 4 shows an illustration of what the final neural network looks like. In summary the input layers have two nodes for each hero id, one for each team. These nodes take the value 1 if the hero is being played on that team and the value 0 if the hero is not played. Some hero-ids are not in use for example hero-id 115 and 116. These are instead used for the match duration and the rank of the game. The three hidden layers have 172 nodes each making them two thirds the size of the input layer. Lastly the output layer has a single node which takes the value 1 if the network predicts the primary team will win the match and take the value 0 if the network predicts the primary team will lose the match.

1. **Model evaluation**

In order to evaluate the model, the value of the output node needs to be discussed. As stated earlier a value close to 1 indicates that the model predicts a win and a value close to 0 predicts a loss. However, this raises the question how a value of 0.5 which is close to the is meant to be interpreted. In that scenario the network does not predict strongly in either way. Regardless of this, the accuracy of the model is defined as the percentage of predictions that are correct when all predictions that are equal to or greater than 0.5 is interpreted as a predicted win and all predictions less than 0.5 is interpreted as a predicted loss.

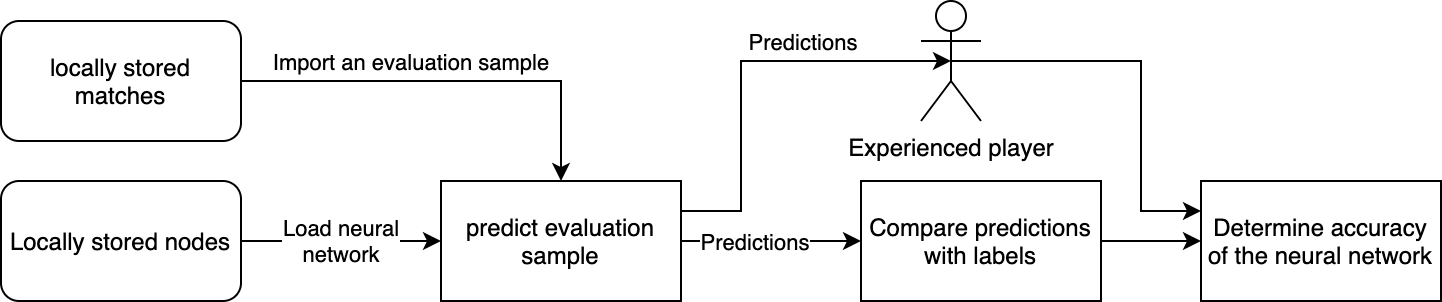


Figure 5 Model Evaluation Architecture

Figure 5 illustrates the way that the model was evaluated. It is to note that different samples were used in training and evaluation. The evaluation sample consisted of about 100000 samples that were randomly selected and excluded from the training data. This data amount should be sufficient to give a good estimate of the accuracy of the model.

## Results and Evaluation

In order to evaluate the results, the value

# Conclusions

Describe the conclusions (reflect on the introduction, background and problem)

Discuss the positive effects and the drawbacks.

# References

References should be numbered using the conventional scientific scheme, using number in square brackets embedded in the text [1]. For those using a reference manager, such as Zotero, please import and employ the citation style, *Journal of Scientific Computing*.

For internet references, where possible reference the webpage author in the same way as a standard reference. If no author is credited, then reference the main webpage (see examples below). For all internet references give the title of the web article, the full web link *and* the date the web link was accessed.

Use one style (preferably Oxford) and be consistent. See IEEE Editorial Style Manual.

IEEE Editorial Style Manual:

<http://www.ieee.org/documents/stylemanual.pdf>

Appendix A

**1. Guidelines**

Guidelines are taken from Procedia.

If you like to use numbered list and sub-headings and columns, use following style:

**1.1. Numbered List**

* 1. A numbered list looks like this
  2. It uses the style ‘List Numbered’
  3. Numbered lists can also be tab indented if necessary 1

**1.2. Sub-headings and columns**

Use only one or two levels of heading. The paper should be in single column format throughout (i.e. not two-column style as required by, for example, the IEEE) all sections in the Paper.

**1.3. Page Start and End**

Do not end a page with a heading or sub-heading and avoid starting a page with an incomplete line. There is no need to include page numbers or running heads; this will be done at our end.

**1.4. Quotations**

Quotations should be embedded within text as follows:

‘Always use single quotation marks (style ‘Quotations’). This is an example of how to cite a quotation that is longer than two lines. For shorter quotations, embed in the normal flow of text, but for two lines or longer use this style with the indentation. State the source page number. Quotations can be edited through use of ... three dots.’ (p. 13)

**2. Diagrams and Figures**

Where possible, enclose diagrams and figure at appropriate points in the main text, rather than appending them at the end of the paper.

**2.1. Tables**

See below for an example table:

Table 1. An example table and caption (style: Figure/Table Caption). Table rows should not cross a page break where possible.

If tables are used, refer to the table by Table 1 before the actual table is presented.

Figures and Tables are numbered independently.

**2.2. Figures (including Diagrams and Graphs)**

Figures should be numbered in sequence, Figure 1, Figure 2 etc. Each Figure should have a brief explanatory title. The text should refer to each table and explain its significance. Electronic images should be provided at an adequately high resolution.

Figure 1. An example figure and caption (style: Figure/Table Caption). All figures should be centered.

If figures are used, write Figure 1 in text to refer to the figure.

**2.3 Footnotes**

Use of footnotes should be avoided. Important and necessary references (including internet links to referenced diagrams and further background details) should be included in the final references section.

|  |  |
| --- | --- |
| Row 1 | Tables should be numbered in sequence, Table 1, Table 2 etc. |
| Row 2 | page3image10093888  Each table should have a brief explanatory title |
| Row 3 | page3image10132800  The text should refer to each table and explain its significance  page3image21048144 |