Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Erik Vabø Vatsvåg

Analysis of injury time in a football game using machine learning techniques

Master's thesis in Cybernetics and Robotics
Supervisor: Ole Morten Aamo
June 2021

NTNU
Norwegian University of
Science and Technology

Erik Vabø Vatsvåg

Analysis of injury time in a football game using machine learning techniques

Master's thesis in Cybernetics and Robotics
Supervisor: Ole Morten Aamo
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis describes analysis done to investigate predictions on injury time in a football game. Statistical and several machine learning techniques have been applied to predict how many minutes will be added by the referee at the end of each half. This research has been done in cooperation with a company called Smartodds, who provides statistical research and sport modeling services for a betting syndicate. The thesis consists of a literature review of football modeling, description of the methods applied, handling and assessment of dataset, provided by Smartodds, results and comparison of the models, and a discussion of the results and a conclusion. Four different models have been developed, a linear model, a Poisson model, a negative binomial model and an artificial neural network model.The performances of the models are compared, and there is not much separating one from another, in the end all of the models are rejected by a 2 goodness of fit test. By a variety of reasons it might be impossible to achieve accurate point predictions of injury time.This can be caused by the incompleteness in the dataset or simple a non-recurrent behavior of the data making it impossible to predict with sufficient confidence based upon neither statistical methods nor machine learning techniques

Contents

List of Figures

List of Tables

Preface

This master thesis has been carried out at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology and in cooperation with a company called Smartodds from January to June 2021. I would like to thank my supervisor, Ole Morten Aamo, for his guidance, and good discussions around the methods and techniques throughout this work. A big thanks to Paul Wikramaratna, my contact person at Smartodds, for tips along the way. Finally I would like to thank my dad for consulting me.

Chapter 1

Introduction

1.1 Introduction

Football is reckoned as the most popular sport in the world. There are many players, about 265 million, and more than 3 billion watching football games FIFA 2021. It has also become a big business where many football clubs being worth more than a billion-dollar and a conglomerate of enterprises such as transfer markets of players, fan clubs, outlet sales of football clothing and equipment, costly rights for the media industry (and expensive pay media channels), newspapers, etc. and a growing betting industry covering everybody from amateurs to professionals and bookmakers.

Today it is possible to bet, not only on the number of goals, name of players scoring but on nearly everything. This thesis is written in co-operation with Smartodds to explore machine learning as a prediction tool. Smartodds is a company that provides statistical research and sport modeling services for a betting syndicate. The consulting consists of mainly giving betting tips with better odds than the bookmakers Biermann 2019.

SmartOdds and gamblers are always on the lookout to gain an advantage over the bookmakers. Professional betters want to secure as high a return of investment as possible. A way to beat the bookmaker is value bets, where the odds given from the bookmaker are higher than the true underlying probability of the given outcome. If the payout is higher than the corresponding probability, the bet will positively return investment over time Trademates 2021. The odds from the bookmaker are based on its analysis and the market; when people place bets, the odds change. There are two prerequisites to find value bets. The first is to have a football model that predicts more accurately than the bookmakers'. The second is that the probability calculated by the model must be lower than the odds provided by the model bookmaker. The objective of this thesis aims to develop models that predict injury time more accurately than bookmakers, such that SmartOdds can provide value bets on injury time.

In the thesis, the task is to use machine learning techniques and statistics to predict the added injury time in football games. The dataset used for the analyses consists of games from the two top divisions in five countries throughout five seasons. Smartodds have provided the datasets.

A football game lasts for 90 minutes, divided into two halves. For each half, the referee can extend each of them with additional time, called injury time, due to various reasons at his discretion. FIFA, the football organization, is developing and organizing football worldwide and has developed rules to ensure that the matches are played with the same rules. The FIFA law 7 is about the game's duration and provides a guiding rule that deals with events in a football game that should qualify for increasing the 45 minutes halves with additional injury time. The allowance for lost time comprises substitutions, assessment, and removal of injured players, wasting time, disciplinary sanctions, medical stoppages, and other causes such as goal celebrations Board 2021. However, it seems to be a common opinion that the referee typically adds 30 seconds of injury time for each

goal and each substitution.

At the end of each half, the 4th referee holds up a sign indicating how many additional minutes should be played, compensating for the lost time. In general, there is a distinct difference in the injury time added between the first and second half. Therefore, the prediction models have been made separately for each half.

The thesis consists of a literature study of machine learning football models, description of the theory, and methodology applied for the predictions, description, handling, and assessment of the datasets provided by Smartodds, description of software tools and programming techniques developed for the analyses, presentation, and comparison of the results of the various models, discussions of the results, conclusions and recommendations for future work.

No analysis on injury time has been found in the literature, so all methods applied in this thesis will be new. This study aims to predict injury time successfully. Nobody knows how injury time is interpreted, so this is an exciting and challenging subject to explore.

1.2 Literature review

1.2.1 Laws of the game

Modern football originated in Britain in the 19th century. It was taken up as winter games at public schools, and The International Football Association Board (The IFAB) introduced some 'universal rules in 1863. IFAB was founded by the four British football associations (The FA, Scottish FA, FA of Wales, and Irish FA) as the worldwide body with sole responsibility for developing and preserving the Laws of the Game. FIFA joined The IFAB in 1913. There are 17 different laws, and each law has its subject. The laws are extensive, and they describe rules to make a football game feasible, including how the game shall be played, the type and size of pitches, the ball, and other equipment. The rules also specify how a free kick, throw-in, corner, and goal-kick must be taken. Law 7 is named "the Duration of the Match" and explains the duration of each half, what allows for additional time, and the duration of potential extra periods. The law consists of five items, of which the third is about the allowance for time lost. The law states that the referee should include additional time for all time lost during a half to:

- substitutions
- assessment and/or removal of injured players
- wasting time
- disciplinary sanctions
- medical stoppages permitted by competition rules, e.g. 'drinks' breaks (which should not exceed one minute) and 'cooling' breaks (ninety seconds to three minutes)
- delays relating to VAR 'checks' and 'reviews'
- any other cause, including any significant delay to a restart (e.g. goal celebrations)

All these events contribute to how much time will be added. At the end of each half, the fourth official holds up a board indicating minimum injury time, and the referee, at his discretion, can extend this. This allows the referee to freely decide what is considered as wasting time and significant delays. As it is not clear how much time should be added for the various causes and it is up to the individual referees, it is not straightforward to estimate the injury time Board 2021.

1.2.2 Injury time

According to research done by Vatsvaag 2020, the events that impact the added time seem to be; goals, substitutions, and other causes that include a significant delay. However, the study was done on a dataset without sanctions. These findings correspond to the FIFA Law 7, concluding that more frequent events, such as free kicks and corners, do not impact the injury time. Such events, however, do affect how much time the ball is in play. Research done by Trainor 2021 shows that in the top five leagues (Premier League, Bundesliga, Ligue 1, Serie A and Primera division), the ball is in play between 52 and 57 minutes median values. Therefore, about 35 minutes in each game, the ball is out of play due to quick events. Even though there is not much time spent on each event, the accumulated time can be pretty high. However, these events are not accounted for in the declared injury time, stated by FIFA law 7. According to Bunnell 2021 free kicks, throw-ins, goal kicks, and corners are the events where most time passes without the ball in play. These discoveries are based on the 2018 World Cup, and Bunnell 2021 timed every game and situation. Hence, a team can waste time during these events without being accounted for in the declared injury time.

Previous research documents home advantage in football Garicano et al. 2005; Lago-Peñas and Gómez-López 2016; Pollard 2006. There are several reasons behind this home advantage, and one is crowd noise. Nevill et al. 2002 discovered that referees viewing challenges with crowd noise awarded 15.5% fewer fouls against the home team than those watching the same challenges in silence. Hence, crowd noise causes referee bias. This bias affects how much injury time is declared, and according to Garicano et al. 2005 in close games, there is a tendency to add more time when the home team is behind and add less time when the home team is leading. This bias increases with crowd size, differences in team abilities and the importance of the game, which is documented by Clarke and Norman 1995; Lago-Peñas and Gómez-López 2016. The importance of the game describes the reward of winning the game. A Champions League game is more important than a Carabao cup game. During the COVID-19 pandemic, there were games in Europe without crowds, and research done by Correia-Oliveira and Andrade-Souza 2021; Konaka 2021; McCarrick et al. 2020 shows that in the home advantage and referee bias decreased without crowds.

1.2.3 Football modeling

Many researchers have tried to predict sports using statistical models. For example, a book called *Moneyball: The Art of Winning an Unfair Game* written by Lewis 2004 and later a movie is about how Billy Beane, the general manager of the Oakland Athletics, started using statistical models to predict individual player performances and using this model to find undervalued players. Moneyball is the most famous story about sports modeling and how a low-budget team could win the Major Baseball League using sabermetric principles. Matthew Benham is the football version of Moneyball. He is called a football scientist and uses statistical models to gain an edge over the bookmakers. Benham does this through his company Smartodds, adviser to a betting syndicate Biermann 2019. Past research about predicting football games can be divided into three parts, statistical models, machine learning models, and rating systems. In this section, statistical models and machine learning models will be reviewed.

Statistical models

Moroney 1975 was one of the first to predict goals in a football game, and he suggested a Poisson model with minor adjustments. These adjustments, in reality, created a negative binomial model, and Moroney argued that it was a good fit for goals scored. He said a negative binomial model would better suit goals scored than a Poisson model because it includes variable means in different games. Further research done by Reep et al. 1971 confirms that a negative binomial model can be applied to football scores . In addition, Reep et al. 1971 argued that a negative binomial model could change mean within games and between games, making it more robust than a Poisson model.

The equation for the goals scored by the negative binomial model is given by:

$$\pi(x_{s,\tau,T_H,T_A}) = \frac{\lambda^{x_{s,\tau,T_H,T_A}}}{x_{s,\tau,T_H,T_A}!}\frac{1}{(1+\frac{\lambda}{w})^w}\frac{\Gamma(w+x_{s,\tau,T_H,T_A})}{\Gamma(w)(w+\lambda)^{x_{s,\tau,T_H,T_A}}} \tag{1.1}$$

where $\lambda = E(x_{s,\tau,T_H,T_A})$ and $w = \frac{\lambda^2}{Var(x_{s,\tau,T_H,T_A})} - \lambda)$, the equations are fetched from Salvesen 2011.

Later academic studies showed that Poisson models are efficient models to predict, and Maher 1982 introduced an independent game-specific Poisson model. This model includes parameters such as each teams' attacking and defensive strengths. The model assumes that in a game between team i and j at home and away respectively, and the score is $(x_{i,j}, y_{i,j})$, $X_{i,j}$ is Poisson distributed with mean $\alpha_i, \beta_j, \gamma$ and $Y_{i,j}$ is Poisson distributed with mean $\alpha_j\beta_i$. Where $\alpha_{i,j}$ and $\beta_{i,j}$ represent the quality of a given team in attack and defense, respectively, and γ represents a home ground advantage that is equal for all teams. After using χ^2 goodness of fit tests, this gives a reasonably good fit to the data and is only rejected in five out of twenty-four games. Maher 1982 initially suggested a more detailed model with separate home and away qualities of attack and defense but realized that the simplification of adding home advantage was sufficient.

The original Poisson model Maher 1982 proposed had some problem with goal differences; hence a bivariate Poisson model was suggested as an extension. Unfortunately, this model was neither flawless, and Dixon and Coles 1997 discovered that it was unable to represent the departure from independence for low scoring games. Following modifications was suggested to the model:

$$Pr(X_{i,j} = x, Y_{i,j} = y) = \tau_{\lambda,\mu}(x,y)Poisson(x|\lambda)Poisson(y|\mu) \tag{1.2}$$

where

$$\begin{aligned} \lambda &= \alpha_i\beta_j\gamma \\ \mu &= \alpha_j\beta_i \end{aligned} \tag{1.3}$$

and

$$\tau_{\lambda,\mu}(x,y) = \begin{cases} 1-\lambda\mu\rho & \text{if } x=y=0, \\ 1+\lambda\rho & \text{if } x=0, y=1, \\ 1+\mu\rho & \text{if } x=1, y=0, \\ 1-\rho & \text{if } x=y=1, \\ 1 & \text{otherwise} \end{cases}$$

and

$$max(-1/\lambda, -1/\mu) \leq \rho \leq min(1/\lambda\mu, 1) \tag{1.4}$$

Later academic studies have also supported Poisson models; among them are Karlis and Ntzoufras 2003 and Angelini and De Angelis 2017. Karlis and Ntzoufras 2003 proposed a bivariate Poisson model with extensions to remove the independence between scores. The extension is to increase the draw probability in games where a draw is more likely. To do this, a diagonal inflated Poisson model was suggested. It is not possible to increase the probability of a draw in models such as double-Poisson models or bivariate Poisson models. The reason Karlis and Ntzoufras 2003 proposed the diagonal inflated Poisson model is because they wanted to predict Serie A in the 1991-1992 season, and in this season, wins were awarded 2 points and draw 1 point. Hence, it is natural that teams want to risk less in order to win, and more games will end in a draw.

Angelini and De Angelis 2017 introduced a Poisson autogression with exogenous covariates (PARX)

model to predict football games. The goal scoring model can be specified as:

$$\begin{aligned} y_t|\mathcal{F}_{t-1} &\sim Pois(\lambda_t), \quad t = 1, ..., T, \\ \lambda_t &= \omega + \sum_{j=1}^{p} \alpha_j \lambda_{t-j} + \sum_{j=1}^{q} \beta_j y_{t-j} + \gamma \boldsymbol{x}_{t-1} \end{aligned} \tag{1.5}$$

$\boldsymbol{x}_{t-1}$ is the vector containing exogenous covariates. The exogenous covariates include information regarding the quality of a football team, such as quality in attack and defense and the team's current form. The expected value for the number of goals are:

$$E[y_t] = E[\lambda_t] = \frac{\omega + E[x_{t-1}]}{1 - \sum_{j=1}^{max(p,q)} (\alpha_j + \beta_j} \tag{1.6}$$

Together with a betting strategy, this model beat the bookmakers, and a χ^2 test failed to reject the null hypothesis of independence between home and away goals.

Machine learning models

The most common machine learning techniques to predict football games are Bayesian methods and neural networks. According to Cheng et al. 2003 a neural network model is more accurate and gives better predictions when compared to statistical models. Furthermore, several academic studies support the use of neural network models when predicting football game outcomes; some researchers that have achieved good results are Arabzad et al. 2014; Cheng et al. 2003; Huang and Chang 2010; Nyquist and Pettersson 2017.

Cheng et al. 2003 built an artificial neural network to forecast the outcome of a football game. The neural network is a classification network consisting of three smaller back-propagation networks and a learning vector quantization method. The LVQ decides which of the BP network should be used based on the strength of the teams. Thus, there are three BP networks, stronger BP, matchable BP, and weaker BP. The output from the network is two predictions, one prediction from the home team's perspective and one prediction from the away team's perspective. These predictions merged into one final prediction. The model predicted Serie A games, and the accuracy of the ANN is measured and compared to other models. The ANN outperforms an Elo model and a ratio model.

The academic studies of Huang and Chang 2010 included a neural network method to predict the winning rate of two teams based on their last stage in the 2006 World Cup. They built a multi-layer perception (MLP) with back-propagation learning. After optimization of the network, an 8-11-1 MLP yielded the best results. However, the network had problems predicting draws. Hence these games are removed from the test set. After the removal of ties, the model has more accurate predictions than football forecasters.

Arabzad et al. 2014 researched the use of artificial neural networks to predict the Iran Pro League. The model is trained on data from the last seven seasons and then tested on the eight final games in the IPL 2013-2014 season. The inputs to the model are; the teams, condition of teams in recent weeks, condition of teams in the league, and quality of opponents in the last games. The MLP consists of two hidden layers with 20 neurons in each and two output layers, home and away goals. This model, similar to the MLP created by Huang and Chang 2010, did not predict a draw. However, the model predicted the correct winner in five games, the proper goal difference in one game, and the correct result.

Nyquist and Pettersson 2017 developed a deep learning method recurrent neural network to predict the outcome of a football match. This model is used to estimate the score in real-time. Naturally, the accuracy of the prediction increases over time and is 0.97 at full-time. However, before the game has started, when only team lineups are known, the accuracy is about 0.44. The model consists of 10 inputs: period, home team, away team, main player, assisting player, position, goal type, card type, penalty type, and substitutes. Successive layers in the model are a one-hot encoding

layer, an embedding layer, and a concatenated layer. These layers transform the inputs (which can be strings) into a vector. The vectors are concatenated and then fed through the network. The network consists of a variable amount of Long Short-Term Memory units. The LSTM units remember values for a while. Finally, the network's output results for the game, either a home win, draw or away win.

Not all machine learning methods within football modeling have been neural networks. Constantinou et al. 2012 used a bayesian network model to forecast association football match outcomes. The model is called 'pi-football and has four inputs for both the home and away team: strength, form, psychology, and fatigue. This model successfully beat the bookmakers' odds over time. However, this model is critically dependent on the Bayesian network structure and the quality of the inputs. The inputs are subjective; hence an expert must assign them accurately. Constantinou 2019 also developed another model to predict football match outcomes; this is called Dolores. Dolores is based on two different techniques: dynamic ratings and hybrid Bayesian networks. It was developed for a competition called Machine Learning for Soccer and finished second. This model was trained on a dataset containing data from 52 football leagues from all over the world. Then the model should predict 206 future games from 26 different leagues. The model makes a good prediction on the new unseen data, meaning it generalizes well, and data from various leagues can improve predictions in a given league.

Chapter 2

Method

In this thesis, several machine learning models have been applied to predict injury time in football games. The different methods are described in this section. Injury time has different mean values and variances in each of the two halves. Hence the models are therefore built separately for each half. It is usually more minutes added in the second half compared to the first one.

The approach to predict the number of injury minutes consists of two parts: firstly, before the game has started, and secondly, during the game in real-time, every 5th minute. For the real-time predictions, the models will be updated every five minutes throughout the games and up to 49 minutes in the first half and up to 94 minutes in the second half, bearing in mind that the second half always starts at 45 minutes. The models are trying to predict how many minutes will be declared of minimum injury time; however, the prediction made on the last time step in each half is a prediction of how long before the referee decides to end the game.

The models that have been developed are:

- Linear regression
- Poisson regression
- Negative binomial regression
- Artificial neural networks

The linear, Poisson and negative binomial models are game-dependent. Based on the previous research, these models are mostly used in modeling football. Hence they are worth exploring and might provide satisfying results.

2.1 Linear regression

A linear model is a model where the output is a fitted linear equation of the regressors. There are different methods to fit the regressors to the response. The technique used in this thesis is the ordinary least squares. Ordinary least squares minimize the sum of squared residuals and do not depend on knowing the distribution of errors. OLS produces the best linear unbiased estimates. The OLS linear model makes two assumptions:

- The response variables, $y_i's$, are independent.
- All observations have the same variance, $\sigma_i^2 = \sigma^2$.

The standard equation for a linear model is:

$$y = \beta \mathbf{x} \tag{2.1}$$

From this model, the equation that defines the estimators is minimizing the sum of squared residuals. The equation for squared residuals is:

$$S = \sum E_i^2 = (\mathbf{y} - \beta \mathbf{x})^T (\mathbf{y} - \beta \mathbf{x}) \tag{2.2}$$

The coefficients are found when minimizing equation (2.2), hence the partial derivative with respect to the coefficients must be taken:

$$\begin{aligned} \left.\frac{\partial S}{\partial \beta}\right|_{\hat{\beta}} &= -2\mathbf{x}^T\mathbf{y} + 2\mathbf{x}^T\mathbf{x}\hat{\beta} = \mathbf{0} \\ \hat{\beta} &= (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \end{aligned} \tag{2.3}$$

assuming the inverse matrix $(\mathbf{x}^T\mathbf{x})^{-1}$ exists, which is true if $\mathbf{x}$ are linearly independent. This results in a fitted regression model;

$$\hat{\mathbf{y}} = \hat{\beta}^T \mathbf{x} \tag{2.4}$$

Equations (2.4) and (2.5) are fetched from Montgomery, Peck and G. Geoffrey Vining 2012; Weiberg 2014.

2.1.1 Before the game

This linear model is a game-dependent model, meaning that every game has an individual linear model. The inputs before the game are home team, away team, referee, and league. The linear model equation in a game k can be written:

$$\hat{y}_k = \hat{\beta}0 + \hat{\beta}1_{i(k)} x1_{i(k)} + \hat{\beta}2_{j(k)} x2_{j(k)} + \hat{\beta}3_{r(k)} x3_{r(k)} + \hat{\beta}4_{r(k)} x4_{r(k)} \tag{2.5}$$

An explanation of this equation:

- y_k is declared injury time in game k.
- $\hat{\beta}0$ is the estimated intercept.
- $i(k) \in \{1, ..., n_{teams}\}$ is an index referring to the home team in game k.
- $j(k) \in \{1, ..., n_{teams}\}$ is an index referring to the away team in game k.
- $r(k) \in \{1, ..., n_{referees}\}$ is an index referring to the referee in game k.
- $l(k) \in \{1, ..., n_{leagues}\}$ is an index referring to the league the game k is played in.

The teams, referees and leagues are being factored as variables and the factors $(\hat{\beta}1_1, .., \hat{\beta}1_{n_{teams}})$,$(\hat{\beta}2_1, .., \hat{\beta}2_{n_{teams}})$,$(\hat{\beta}3_1$, $(\hat{\beta}4_1, ..., \hat{\beta}4_{n_{leagues}})$ are vectors of coefficients which are to be estimated. While $x0 = 1$, $(x1_1, .., x1_{n_{teams}})$,$(x2_1, .., x2_{n_{team}}$ $(x4_1, ..., x4_{n_{leagues}})$ are the inputs in a specific game. The coefficients and intercept are estimated by minimizing equation (2.3).

2.1.2 Real time prediction

For the real time predictions the input variables to the model are; pregame prediction, start period, end period, goals, substitutions and total delay seconds. The pregame prediction is made by the model described in section 2.1.1. The linear model equation can be written:

$$\hat{y}_k = \hat{\beta}0 + \hat{\beta}1x1_{pp_{d(k)}} + \hat{\beta}2x2_{sp} + \hat{\beta}3x3_{ep} + \hat{\beta}4x4_{goals} + \hat{\beta}5x5_{subs} + \hat{\beta}6x6_{ds} \tag{2.6}$$

An explanation of this equation:

- y_k is declared injury time in game k.
- β_0 is the intercept.
- $d(k) \in \{1, ..., n_{games}\}$ is an index referring to the pregame prediction in game k.
- $x1_{pp_{d(k)}}$ is the pregame prediction in game k.
- $x2_{sp}$ is the start of the time step.
- $x3_{ep}$ end of the time step.
- $x4_{goals}$ is the number of goals so far in the half.
- $x5_{subs}$ is the number of substitutions so far in the half.
- $x6_{ds}$ is the total delay seconds so far in the half.

$\hat{\beta}$ is a vector with the coefficients, and $\mathbf{x}$ is a vector containing all the inputs. $\mathbf{x}$ can be written like this $\mathbf{x} = [1, pregame\_prediction_k, start\_period, end\_period, goals, subs, total\_delay\_seconds]$. The coefficients and intercept are estimated minimizing equation (2.3).

2.1.3 Coefficient of determination

It is common to use the coefficient of determination, or R^2, to evaluate the goodness of fit for linear models. R^2 is a measure of how much of the variance in the response variable is predicted by the linear model. Thus, increasing values of R^2, results in a more linear relationship between the regressors and the response variable. R^2 can be negative; this means that the predictions are worse than predicting the mean every time. The maximum value for R^2 is one, and this represents a perfect positive linear relationship. Equation (2.7) shows how R^2 is calculated, and is fetched from Montgomery, Peck and Geoffrey G. Vining 2006.

$$\begin{aligned} TSS &= \sum E_i'^2 = \sum (Y_i - \bar{Y})^2 \\ RSS &= \sum E_i^2 = \sum (Y_i - \hat{Y}_i)^2 \\ RegSS &= TSS - RSS \\ R^2 &= \frac{RegSS}{TSS} \end{aligned} \tag{2.7}$$

2.2 Poisson regression

The linear model was generalized by Nelder and Wedderburn. Generalized linear models can be used for count data, and according to Colin and Pravin 2013a, a GLM makes two assumptions, the response variable must be in the Exponential family distribution, and the function of the mean must be linear. The exponential family includes all distributions that can be written in the exponential-family form. This includes both the Poisson and NB distribution. The equations for Poisson and

NB distribution written in the exponential-family form are shown in equations (2.8) and (2.13). The function of the mean is called the link function, and there are different link functions. Poisson models and NB models use the log function as a link function, shown in equation (2.9). GLM uses maximum likelihood to estimate the coefficients. The maximum likelihood equation is derived from the exponential family, and equation (2.10) shows the log-likelihood equation for a Poisson regression model. The process described is applied two times separately, one for each half. Hence there are two separate models, each with its own set of vectors with estimated values.

$$f(y, \lambda) = exp\{yln(\lambda) - \lambda - ln\Gamma(y+1)\} \tag{2.8}$$

$$ln(\lambda_i) = x\beta \tag{2.9}$$

$$\mathcal{L}(x\beta; y) = \sum_{i=1}^{n} \{y_i(x_i\beta) - exp(x_i\beta) - ln\Gamma(y_i+1)\} \tag{2.10}$$

The log-likelihood function is maximized by taking the partial derivatives with respect to β, the coefficients, and solving the equation for each input. The mean and variance of the Poisson model is λ. Equations (2.8) to (2.10) are taken from Hardin and J. W. Hilbe 2012.

In this thesis, a Poisson model is developed to predict injury time. The Poisson distribution is only valid for nonnegative integers, which is very suitable for predicting counting data. Multiple researchers have received good results using Poisson regression to model goals in football, and it is the most used count regression. Hence, it is natural to see if this regression can provide satisfying results.

2.2.1 Before game

Using equations (2.8) to (2.10) with the inputs equal as with the pregame linear model; home team, away team, referee and league following equations are derived for injury time in game k:

$$\begin{aligned} Y_k &\sim Poisson(\mu_k) \\ P(Y_k = y_k|\lambda_k) &= \frac{\lambda_k^{y_k}}{y_k!} exp^{-\lambda_k} \\ ln(\lambda_k) &= \beta_0 + \beta_{i(k)}x1_{i(k)} + \beta_{j(k)}x2_{j(k)} + \beta_{r(k)}x3_{r(k)} + \beta_{r(k)}x4_{r(k)} \end{aligned} \tag{2.11}$$

where;

- y_k is declared injury time in game k.
- β_0 is the intercept.
- $i(k) \in \{1, ..., n_{teams}\}$ is an index referring to the home team in game k.
- $j(k) \in \{1, ..., n_{teams}\}$ is an index referring to the away team in game k.
- $r(k) \in \{1, ..., n_{referees}\}$ is an index referring to the referee in game k.
- $l(k) \in \{1, ..., n_{leagues}\}$ is an index referring to the league the game k is played in.

The maximum likelihood estimator assumes that the output variable and the log function for the expected value follows equation (2.11) are Poisson distributed. The teams, referees and leagues are being factored as variables and the factors $(\beta 1_1, .., \beta 1_{n_{teams}})$,$(\beta 2_1, .., \beta 2_{n_{teams}})$,$(\beta 3_1, .., \beta 3_{n_{ref}})$, $(\beta 4_1, ..., \beta 4_{n_{leagues}})$ are vectors of coefficients which are to be estimated. While $x0 = 1$, $(x1_1, .., x1_{n_{teams}})$,$(x2_1, .., x2_{n_{team}}$

$(x4_1, ..., x4_{n_{leagues}})$ are the inputs in a specific game, deciding which coefficient in β that is applicable. Maximum likelihood estimates are chosen based on the set of parameters which results in the highest probability to produce the observed data.

The process described above must be done two times separately, one for each half. This results in two separate models, each with its own set of vectors with estimated values.

2.2.2 Real time prediction

For the real-time predictions, the inputs to the model are; pregame prediction, start period, end period, goals, substitutions, and total delay seconds. The probability function for observing a given number of injury time is equal equation (2.11). The link function, on the other hand, has changed with the new inputs. Hence, the new link function can be written:

$$ln(\lambda_k) = \beta_0 + \beta_1 x1_{pp_{d(k)}} + \beta_2 x2_{ep} + \beta_3 x3_{ep} + \beta_4 x4_{goals} + \beta_5 x5_{subs} + \beta_6 x6_{ds} \tag{2.12}$$

An explanation of this equation:

- y_k is declared injury time in game k.
- β_0 is the intercept.
- $d(k) \in \{1, ..., n_{games}\}$ is an index referring to the pregame prediction in game k.
- $x1_{pp_{d(k)}}$ is the pregame prediction in game k.
- $x2_{sp}$ is the start of the time step.
- $x3_{ep}$ end of the time step.
- $x4_{goals}$ is the number of goals so far in the half.
- $x5_{subs}$ is the number of substitutions so far in the half.
- $x6_{ds}$ is the total delay seconds so far in the half.

β is a vector with all the coefficients, $\beta = [\beta 0, \beta 1, \beta 2, \beta 3, \beta 4, \beta 5, \beta 6]$, and $\mathbf{x}$ is a vector containing all the inputs and looks like this $\mathbf{x} = [1, pregame\_prediction_k, start\_period, end\_period, goals, subs, total\_delay\_seconds]$. The coefficients are estimated using maximum likelihood estimator, equation (2.10).

2.3 Negative Binomial regression

Negative binomial regression is a GLM where the response variable is negative binomial distributed. It is the second most used count regression. The most common negative binomial distribution, NB2, is derived from a Poisson-gamma mixture distribution. The Poisson-gamma mixture distribution is suitable if there is overdispersion or underdispersion in the data and it has the shape of a gamma distribution. The shape can take other forms than the gamma distribution, but then the Poisson-gamma mixture should be alternated accordingly. In the case of an unknown shape, statisticians tend to choose the negative binomial distribution derived from the Poisson-gamma mixture distribution. Overdispersion takes place if the variance in a dataset is higher than the mean in the dataset, and if the variance is lower than the mean, there is underdispersion. If there exists dispersion in the data, NB2 is the best model. At the start of football modeling, this was the most used, and it is a more robust distribution than the Poisson distribution due to the variance separates from the mean. This regression will yield satisfying results if injury time is negative binomial distributed. Equation (2.13) includes the negative binomial distribution equations written in exponential-family notation. The model is created using a maximum likelihood estimator, and the log-likelihood function is showed in equation (2.15). There are two ways to derive these

functions, using the Poisson-gamma mixture and Bernoulli trials, where the probability function describes the probability of observing y failures before the rth success. The theory provided is found in Hardin and J. W. Hilbe 2012; J. M. Hilbe 2011. Even though they are alike, the Poisson-gamma mixture is how the equations are derived in this thesis.

$$f(y;\lambda,\alpha) = exp\left\{yln\left(\frac{\alpha\lambda}{1+\alpha\lambda}\right) + \frac{1}{\alpha}ln\left(\frac{1}{1+\alpha\lambda}\right) + ln\Gamma\left(y+\frac{1}{\alpha}\right) - ln\Gamma(y+1) - ln\Gamma\left(\frac{1}{\alpha}\right)\right\} \tag{2.13}$$

$$ln\left(\frac{\alpha\lambda_i}{1+\alpha\lambda_i}\right) = x_i\beta \tag{2.14}$$

$$\mathcal{L}(x\beta;y,\alpha) = \sum_{i=1}^{n}\left\{y_i ln\left(\frac{\alpha exp(x_i\beta)}{1+\alpha exp(x_i\beta)}\right) - \frac{1}{\alpha}ln(1+\alpha exp(x_i\beta)) + ln\Gamma\left(y_i+\frac{1}{\alpha}\right) - ln\Gamma(y:i+1) - ln\Gamma\left(\frac{1}{\alpha}\right)\right\} \tag{2.15}$$

The variance in the negative binomial model is:

$$V(\lambda) = \lambda + \alpha\lambda^2 \tag{2.16}$$

The offset is parametrized to α and α corresponds to the assumed dispersion in the dataset, when $\alpha = 0$, the model is equal to a nested Poisson model. The assumed dispersion increases as α increases. Hence, α are estimated using the maximum likelihood estimator, by maximizing equation (2.15) with respect to *alpha*. Equations (2.13) to (2.16) are fetched from Hardin and J. W. Hilbe 2012; J. M. Hilbe 2011.

2.3.1 Before game

$$\begin{aligned} Y_k &\sim NB(\lambda_k,\alpha) \\ f(Y_k = y_k|\lambda_k,\alpha) &= \frac{\Gamma(y_k+1/\alpha)}{\Gamma(y_k+1)\Gamma(1/\alpha)}\left(\frac{1}{1+\alpha\lambda_k}\right)^{1/\alpha}\left(1-\frac{1}{1+\alpha\lambda_k}\right)^{y_k} \\ ln\left(\frac{\alpha\lambda_k}{1+\alpha\lambda_k}\right) &= \beta_0 + \beta_{i(k)}x_{i(k)} + \beta_{j(k)}x_{j(k)} + \beta_{r(k)}x_{r(k)} + \beta_{r(k)}x_{r(k)} \end{aligned} \tag{2.17}$$

where;

- y_k is declared injury time in game k.
- β_0 is the intercept.
- $i(k) \in \{1,...,n_{teams}\}$ is an index referring to the home team in game k.
- $j(k) \in \{1,...,n_{teams}\}$ is an index referring to the away team in game k.
- $r(k) \in \{1,...,n_{referees}\}$ is an index referring to the referee in game k.
- $l(k) \in \{1,...,n_{leagues}\}$ is an index referring to the league the game k is played in.

The teams, referees and leagues are being factored as variables and the factors $(\beta 1_1,..,\beta 1_{n_{teams}})$,$(\beta 2_1,..,\beta 2_{n_{teams}})$,$(\beta 3_1,$ $(\beta 4_1,...,\beta 4_{n_{leagues}})$ are vectors of coefficients which are to be estimated. While $x0 = 1$, $(x1_1,..,x1_{n_{teams}})$,$(x2_1,..,x2_{n_{team}}$ $(x4_1,...,x4_{n_{leagues}})$ are the inputs, one per home team, away team, referee and league. They are estimated using a maximum likelihood estimator. The reason for developing a Negative Binomial model is because underdispersion occurs in the data. Hence a negative binomial model can

possibly provide better results than the Poisson model. These are the two most common count distributions, hence it is natural to explore these statistical learning methods to see if can yield satisfying results. The process must be done twice, resulting in two separate sets of vectors, one for each half.

2.3.2 Real time prediction

Equally as with the Poisson models, the probability function for observing a given value of injury time is equal in both models. On the other hand, the link function change when the inputs change. The new link function can be written:

$$ln\left(\frac{\alpha\lambda_k}{1+\alpha\lambda_k}\right) = \beta_0 + \beta_1 x1_{pp_{d(k)}} + \beta_2 x2_{ep} + \beta_3 x3_{ep} + \beta_4 x4_{goals} + \beta_5 x5_{subs} + \beta_6 x6_{ds} \tag{2.18}$$

An explanation of this equation:

- y_k is declared injury time in game k.
- β_0 is the intercept.
- $d(k) \in \{1, ..., n_{games}\}$ is an index referring to the pregame prediction in game k.
- $x1_{pp_{d(k)}}$ is the pregame prediction in game k.
- $x2_{sp}$ is the start of the time step.
- $x3_{ep}$ end of the time step.
- $x4_{goals}$ is the number of goals so far in the half.
- $x5_{subs}$ is the number of substitutions so far in the half.
- $x6_{ds}$ is the total delay seconds so far in the half.

β is a vector with all the coefficients, $\beta = [\beta 0, \beta 1, \beta 2, \beta 3, \beta 4, \beta 5, \beta 6]$, and $\mathbf{x}$ is a vector containing all the inputs and looks like this $\mathbf{x} = [1, pregame\_prediction_k, start\_period, end\_period, goals, subs, total\_delay\_seconds]$.

2.4 Artificial neural network

No certainty declared injury time follows a known statistical distribution; if so, a machine learning technique might provide better results. Multi-layer artificial neural networks have a wide area of use due to its flexibility, and it is robust against errors in training data. The neural network is dynamic and is built to achieve the best possible prediction. The ANN outputs a floating number.

An ANN consists of layers of nodes and connections between the nodes. The connections between the nodes consist of multiplication, called weights, and activation functions. During the training of a neural network, an input, $\mathbf{x}$, is fed through the network and produces an output, $\mathbf{y}$ and a scalar cost. This is called forward propagation. The cost is then sent back through the network to compute the gradients, called back-propagation. The gradients are calculated from a chosen loss function. The next step is to use the gradients to train the weights using an optimization algorithm. The goal is to optimize the same loss function. In a regression neural network, it is standard to use mean squared error as a loss function. This is the mean of the sum of squared residuals between the actual values and predicted values. MSE can be used in all regression problems and does not make any assumption about the output variable Goodfellow et al. 2016.

An optimized neural network is achieved through hyperparameter optimization. In Keras, there exist algorithms that tune the hyperparameters automatically, and it is called Keras tuner. There are

multiple Keras tuners, such as random search, a brute-force search algorithm, Bayesian optimization, and Hyperband. According to Snoek et al. 2012 Bayesian optimization finds hyperparameters significantly faster than human experts. Hence BO is used in this thesis. BO uses machine learning to find the set of hyperparameters that yields an optimized fit of the ANN. This is done by creating a Gaussian process model that predicts regions in the hyperparameter space where it most likely increases the model's performance. The GP model makes its prediction based on results from earlier trials with different sets of hyperparameters, and the set that yielded the best result so far is saved. For the next step, the model finds the hyperparameter settings which have the most significant expected improvement. The hyperparameter tuner will decide the number of hidden layers, the number of neurons in each hidden layer, activation function, and optimization function. Additionally, the number of epochs the network is trained.

2.4.1 Before game

An ANN only accepts numeric inputs; however, before the game, the inputs are; the home team, the away team, the referee, and the league of the game. All these inputs are strings; hence some transformation is necessary. Each input will be transformed using the same technique but transformed separately. The first step of the transformation is to use a label encoder, and the label encoder converts a list of words to a list of integers. Each word is assigned one integer, and the range of integers in the new list is equal to the number of unique words in the original list. However, this transformation is not enough, because now if these are sent as inputs to the network, the values are coherent and random, which is unwanted. The next step is to transform the integer values into a vector such that equal inputs will have more equal vectors. Hence, this transformation is trained using an embedding layer for each input. This is done so that, for example, good teams in the same league that often receives the same amount of injury time are grouped. Another advantage of using embedding layers is that the length of the vector is chosen, and a rule of thumb is to reduce the size of the vector to half of the unique words in the input list. Another possible technique is the one hot encoder, which also takes a list of words and transforms each word into a vector. This vector has the length of the unique values in the input list, and each unique word transforms to an index in the vector. This technique is not used in this thesis because the dimensionality can not be reduced, and the transformation is not learned. Other researchers have received good results using embedding layers Bengio et al. 2003; Bordes et al. n.d.; Chollet et al. 2015. After the embedding layer, all the vectors are concatenated into one vector before fed through the network.

Figure 2.1 shows the ANN before the Keras tuner has optimized the hyperparameters.

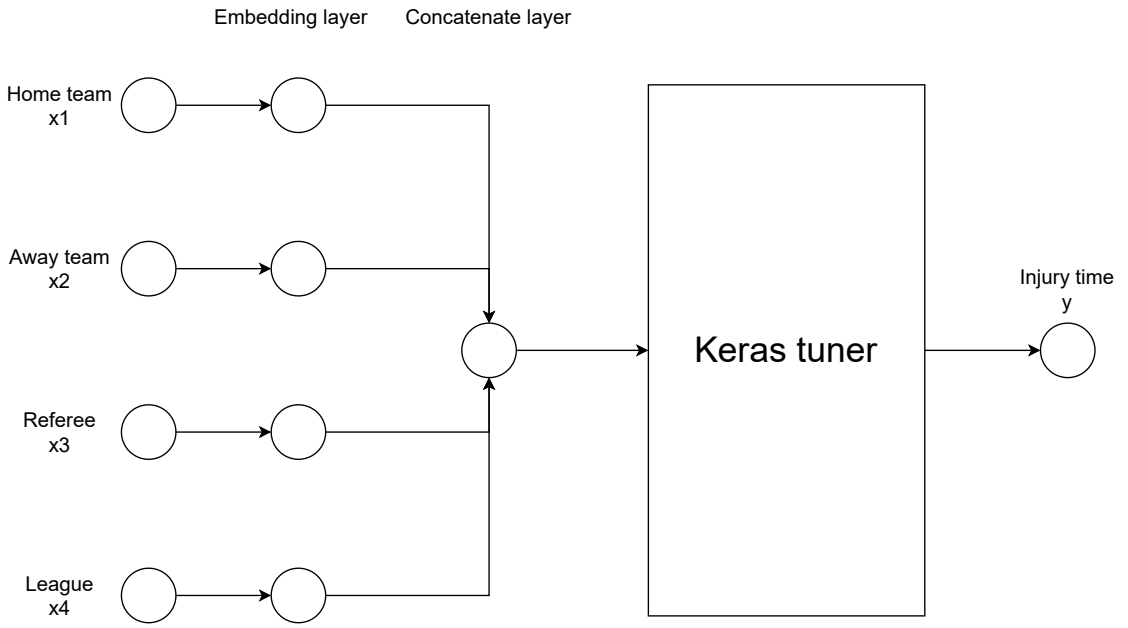


Figure 2.1: ANN before game model Keras tuner has optimized hyperparameters

2.4.2 Real-time prediction

The inputs for the real-time predictions are start period, end period, pregame prediction, goals, substitutions, total delay in seconds, so far in each half. All of these are numeric, hence there is no need for an embedding layer or such. Figure 2.2 shows the ANN before the Keras tuner has optimized the hyperparameters.

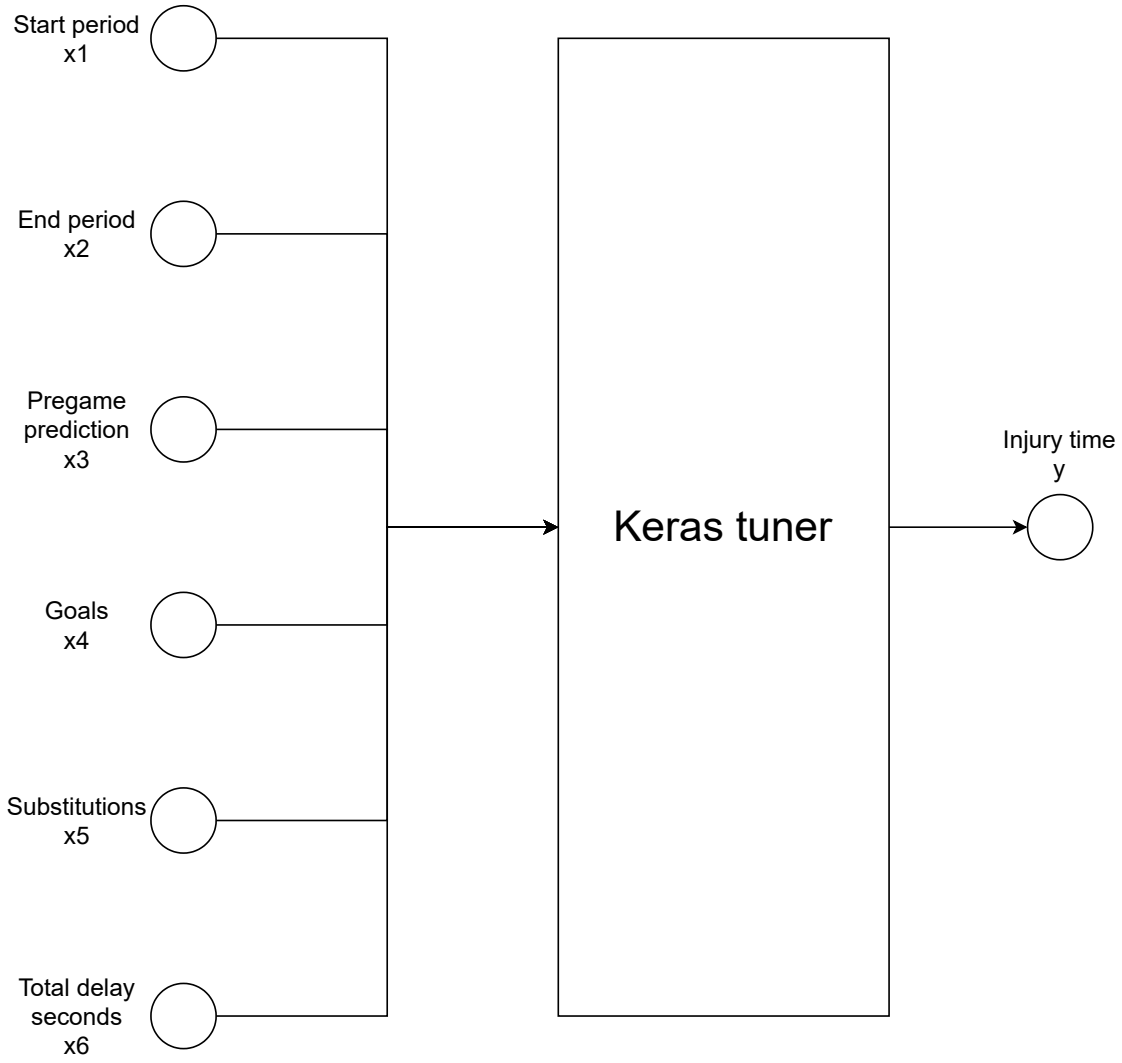


Figure 2.2: ANN real time model Keras tuner has optimized hyperparameters

2.5 Model performance

Each model outputs injury time as a float value, this will be used to calculate errors, and it will be rounded to the closest integer to calculate accuracy. A χ^2 goodness of fit test will be conducted, the linear model and the ANN model will be tested on rounded predictions, while the Poisson model and NB model will be tested on the total sum of expected frequencies. All the models will predict values on unseen data from the test set, and then different kinds of measurements will be taken to analyze the model's performance. The floating number from the Poisson model and NB model is the mean of a corresponding distribution. The linear model and ANN model, on the other hand, do not have a distribution coherent to the number.

The model performances will be evaluated based on error measurements, a goodness of fit test, accuracy with rounded point predictions, and uncertainty in the model described in the following subsections. The actual results from these tests will be further elaborated in the section for results and discussions.

2.5.1 Error measurements

The models' predictions are compared to the actual declared injury times, and different kinds of error measurements are taken and compared to each other. The error measurements are mean squared error, root mean squared error, mean absolute error, mean absolute percentage error, and each is based on the float prediction from the model. Hence, these are point errors. All of the models will be compared to each other. Equation (2.19) shows the error equations.

$$\begin{aligned} MSE &= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \\ RMSE &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \\ MAE &= \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \\ MAPE &= \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| * 100\% \end{aligned} \tag{2.19}$$

where y is the actual declared injury time, $\hat{y}$ is the predicted injury time and N is the number of games in the test set.

2.5.2 Goodness of fit test

A general goodness of fit test for count models is the χ^2 goodness of fit test. This test provides a measure of fit, a χ^2 statistic, calculated using equation (2.20). The test is based on actual counts and predicted counts of injury time for all the different values of injury time. However, if the expected value is zero, which may be true for large values of injury time, the sum is infinity. Hence, the remaining counts are grouped into one cell for low counts to avoid the values getting too small. The numbers are based on rounded predictions for the linear model and the ANN model. While, the Poisson model and NB model, on the other hand, it is based on excepted counts. The outputs from the Poisson model and NB models are the expected value in a given game. The expected value is used in the PMF, probability mass function, to calculate a probability for each frequent number of declared injury time. All the probabilities for each specific minute are summed, and these are the expected counts. The outcome of the test χ^2 goodness of fit test is different based on whether the number is rounded or not. For the linear and ANN regression model, the chi^2 goodness of fit test measures how good the model is to predict injury time correctly. In contrast, the Poisson and NB models, the result is a measure of how well the coherent distributions fit actual declared injury time.

$$\chi^2 = \sum_{i=1}^{N}\frac{(O_i - E_i)}{E_i} \tag{2.20}$$

$$f(x;k) = \frac{x^{k/2-1}exp(-x/2}{2^{k/2}\Gamma\left(\frac{k}{2}\right)} \tag{2.21}$$

Equations (2.20) and (2.21) and explanations are fetched from Colin and Pravin 2013b. The fit decreases for increasing values of χ^2 statistic, meaning there are higher differences between the two distributions. A p-value is calculated, which is a probability of the two sets of data being from equal distributions. The p-value is derived using equation (2.21), which is the probability density function for the χ^2 distribution, where x is the χ^2 statistic and k is the degrees of freedom.

Typical for the first half model, the model is evaluated at one, two, and three minutes count, where the three-minute count is the sum of all predictions above three minutes, same for actual injury times. Then, the degrees of freedom are two. If the calculated p-value is less than 0.05, there is a significant difference between the declared injury times and predicted injury times, and the model should be rejected.

2.5.3 Accuracy

Accuracy is a measure of how often the model predicts correctly, and it is the count of correct predictions divided by the number of total predictions. The predicted value for each model is rounded to the nearest integer and compared to the actual declared injury time. The accuracy is derived by dividing the number of times they are equal by the total number of games.

2.5.4 Uncertainty

The models will not predict correctly every time, and each model has an uncertainty. There are two different methods to incorporate uncertainty, confidence intervals, and prediction interval. Each coefficient estimated in the linear model, Poisson model, and NB model has a confidence interval, the most commonly used value is 95%. Using these coefficients with the confidence interval, the true model of injury time is within the upper and lower limits 95% of the time. The confidence intervals are not applied to the neural network because the inputs are not assigned coefficients. On the other hand, prediction intervals give certainty to the prediction, and in a game, the declared injury time is within the lower and upper limits 95% of the time. In results, each model's standard error on prediction, mean of all estimated coefficients for the home teams, mean of all the estimated coefficients for the away teams, and mean standard error to the all home team and away team coefficients are presented. Equation (2.23) shows how the standard error is derived and how prediction intervals can be derived based on the standard error. T critical value is the cut-off point on the t distribution, found in a T-distribution table Glen 2021, and in this thesis, $t_{crit} = 2.132$ in the first half and $t_{crit} = 1.895$ for the second half. On the other hand, equation (2.22) shows how confidence intervals are calculated. To summarize, the confidence intervals are uncertainty in the regressors coefficients, x, and prediction intervals are uncertainty in the predictions, y.

$$\begin{aligned} \hat{y} \pm t_{crit} s.e. \\ s.e. = s_{yx} \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{SS_x}} \end{aligned} \tag{2.22}$$

Equation (2.23) shows how the prediction interval for a predicted injury time, y_0, in a game x_0.

$$\begin{aligned} \hat{y_0} \pm t_{crit} s.e. \\ s.e. = s_{yx} \sqrt{\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{SS_x}} \end{aligned} \tag{2.23}$$

Equations and explanations are fetched from Zaiontz 2021.

2.5.5 Bar plot

A bar plot showing the counts of predicted minutes of injury time compared to actual declared minutes of injury time will be shown for each model. This is to show the distribution of predictions compared to the distribution of actual injury times. It is helpful to visualize the distributions to see if there is correspondence between the model and injury time. For real-time models, there will be a bar plot of counts for every step; this way, it is possible to see how the model changes its

prediction as time passes in the games. The numbers are based on rounded predictions for the linear model and the ANN model; for the Poisson model and NB model, on the other hand, it is based on excepted counts.

2.5.6 Confusion matrix

For each model, a confusion matrix will be presented. This is a matrix where the predicted injury times are on the horizontal axis and actual declared injury times on the vertical axis. The value inside the matrix is how often a given situation occurs; an example is in place (1,2) is the amount of time the actual declared injury time is one minute, and the model predicts two minutes. The values along the diagonal are the number of games where the model has predicted correctly. The confusion matrices are presented in addition to the bar plots for further investigation and analysis. The confusion matrix allows for evaluating the models at specific predictions. Each confusion matrix is created by rounding the predicted mean to the closest integer.

Chapter 3

Datasets

3.1 Description

The datasets that this research is based upon is provided from Smartodds. There are three different datasets containing information about football games. The datasets cover games from the top 2 leagues in England, Spain, Italy, Germany and France for the seasons starting summer 2014 to those ending summer 2019. This adds up to 17863 different games. One dataset contains all the meta information about the games including kick off datetime, country, season, competition, referee, team names, full time goals for each team. Also it contains information about what happened during regular playing time, excluding events that took place in injury time, including period, goals, corners, free kicks, substitutions, total delay seconds and declared injury time.

The second dataset contains the counts of goals, corners, free kicks and substitutions (across both teams) in each 5 minute period of each half (i.e. these are not cumulative). This is set up so that the start of the first period is everything on the clock from 00:00 to 04:59, and the second period is from 05:00 to 09:59, and so on. This is analogous to the first dataset, but the counts are split into smaller subsets, and it also include the counts of what happened in injury time, unlike the first dataset. All the games in the first dataset are contained inside here.

The third dataset contains details the information about delays in each game: for each fixture the start and end time of each recorded delay in each half is noted and each delay can be uniquely identified by the combination of fixture id - period - delayGroup. These can be used to calculate the length of any delays and identify when they happened.

Instead of using three different datasets the necessary information was merged into one dataset. The first dataset is used to make the pregame predictions, these predictions are included in the second dataset. Next step is to include the delays from the third dataset in the second dataset. For every delay in the third dataset, this must be placed correctly in the second one. This process consists of two steps, first is to find the matching fixture id and the second is to place it in the correct time period. Last step of preparing the dataset is to make the counts of goals, corners, free kicks, and substitutions to cumulative sums so far in each half.

3.1.1 Cleaning

After inspecting the dataset, some errors were discovered. Hence, some cleaning of the dataset was necessary. All games where declared injury time was zero in either one of the halves or both, the value for declared injury time in both halves are N/A. This causes some problems, firstly data is lost in the half that did not have zero minutes of declared injury time, secondly there is no way to tell if one of the halves is missing and which one and finally there might be faults in the dataset, but these are impossible to separate from games with zero minutes of declared injury time. In most of the games there are no record online of declared injury time. Hence, all these games, 5714,

are removed from the dataset and there occur no values of zero.

3.2 Statistics in the dataset

To get a better understanding of the dataset some statistics will be presented. These statistics are based on the cleaned dataset. The mean declared injury time for first period is 1 minute and 43 seconds, and for the second period, 3 minutes an 44 seconds, and the variance in injury time in first half is 1.00, and the second half, the variance is 1.30. The variances in both halves are less the mean in the respective half. Hence, there is underdispersion in the dataset, meaning the variation is smaller than the mean, in the data. In Figures 3.1, 3.2, the distribution of games-declared injury time are shown. Regarding first half, more than half of the games have one minute of declared injury time and regarding second half, more than half of the games have either three or four minutes of declared injury time.

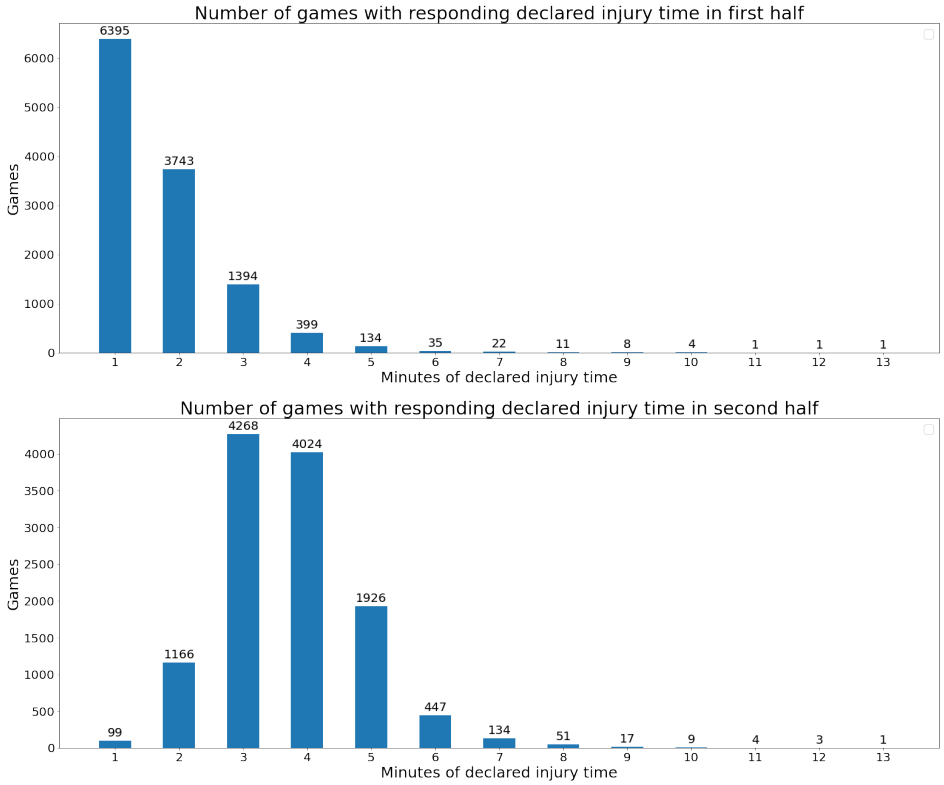


Figure 3.1: The number of games with corresponding declared injury time

Some further investigations of the dataset have been done, to obtain a better understanding of the variability of the declared injury time between different leagues, any seasonal changes, variability due to different referees disregarding referees with less than 25 games, and potentially differences between all the teams. All these comparisons are plotted and presented in figures 3.3 to 3.6 respectively. The results of these comparisons and statistics in the dataset will be further discussed in the discussion.

Figure 3.3 shows the variability between the different leagues. The highest values in the first half are in the English premiership with a mean value in excess of 2 minutes. The Spanish league has the lowest average less than 1 and a half minute, giving a difference of about 45 seconds. In

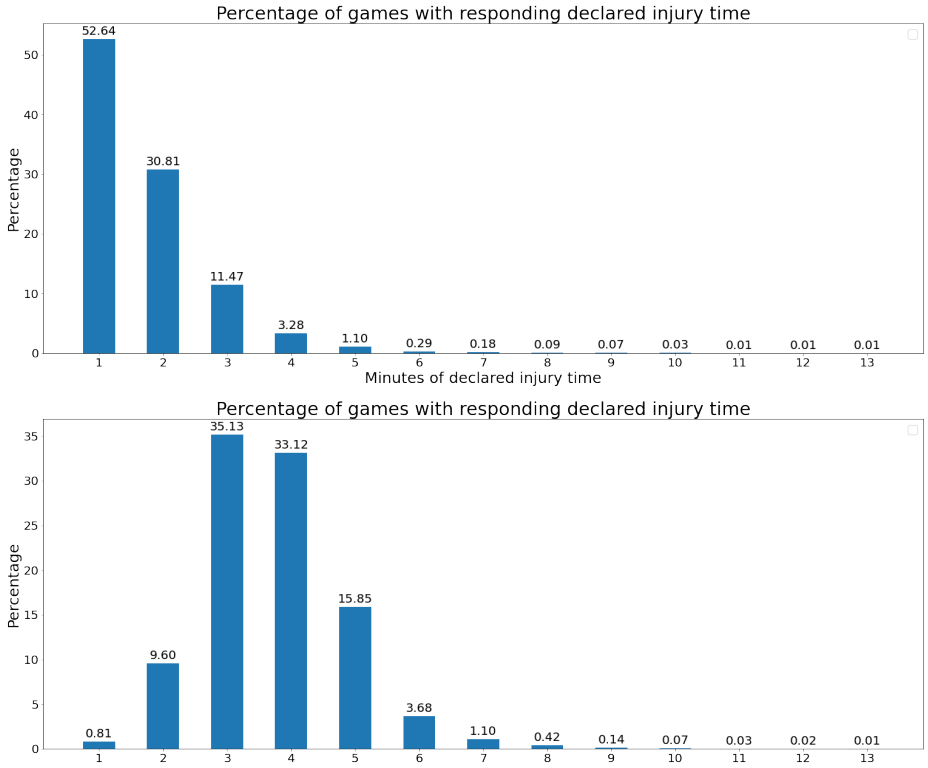


Figure 3.2: The percentage of games with corresponding declared injury time in the dataset

second half, The English premiership is still high, but the Italian B league is higher with a time about 4.5 minutes and the German 2nd league is showing the lowest value, just above 3 minutes. However, there are less games in the Italian B league and the German 2nd league, due to corrupt data, meaning that there might have been more games with zero minutes of injury time. Hence, the mean would have been lower. It can be concluded that it seems that the rules are practicing very similar between the countries and in general maybe a slightly higher injury time in the lowest divisions.

In figure 3.4 the variation in the different seasons is shown. It seems like there is a trend that the injury time has increased steadily, although very slowly over the 5 year period, about 20 seconds in the first half and 25 seconds for second half.

Figures 3.5 and 3.6 show the variation between the referees and the different teams. The distributions seem to be within the same variability as the other data.

The variability in number of games between the referees is shown in figure 3.7 below. The highest number of games for a referee is up to 300 but most of the referees have games in the region of 25 to 100 games.

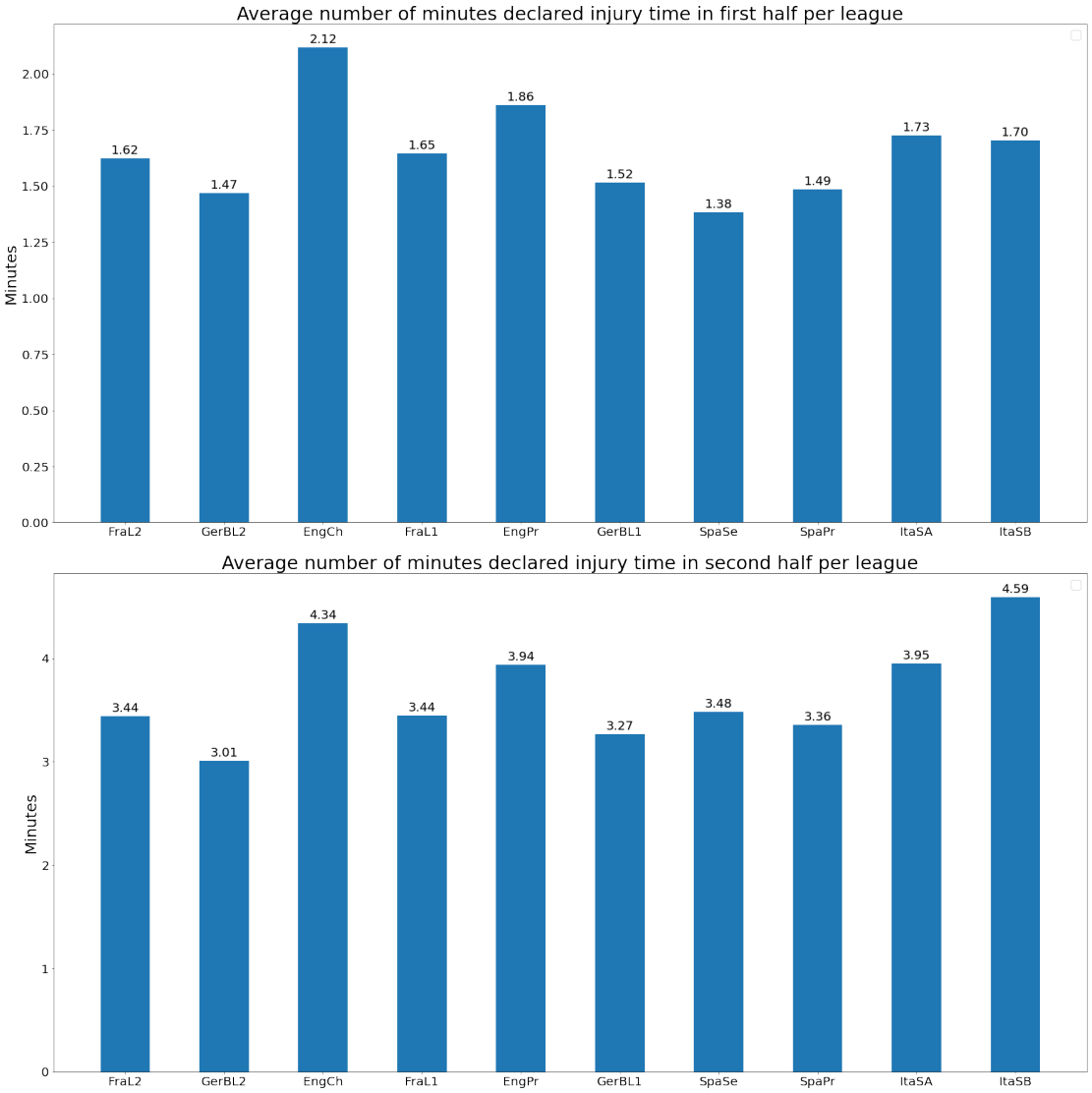


Figure 3.3: Average declared injury time in each league

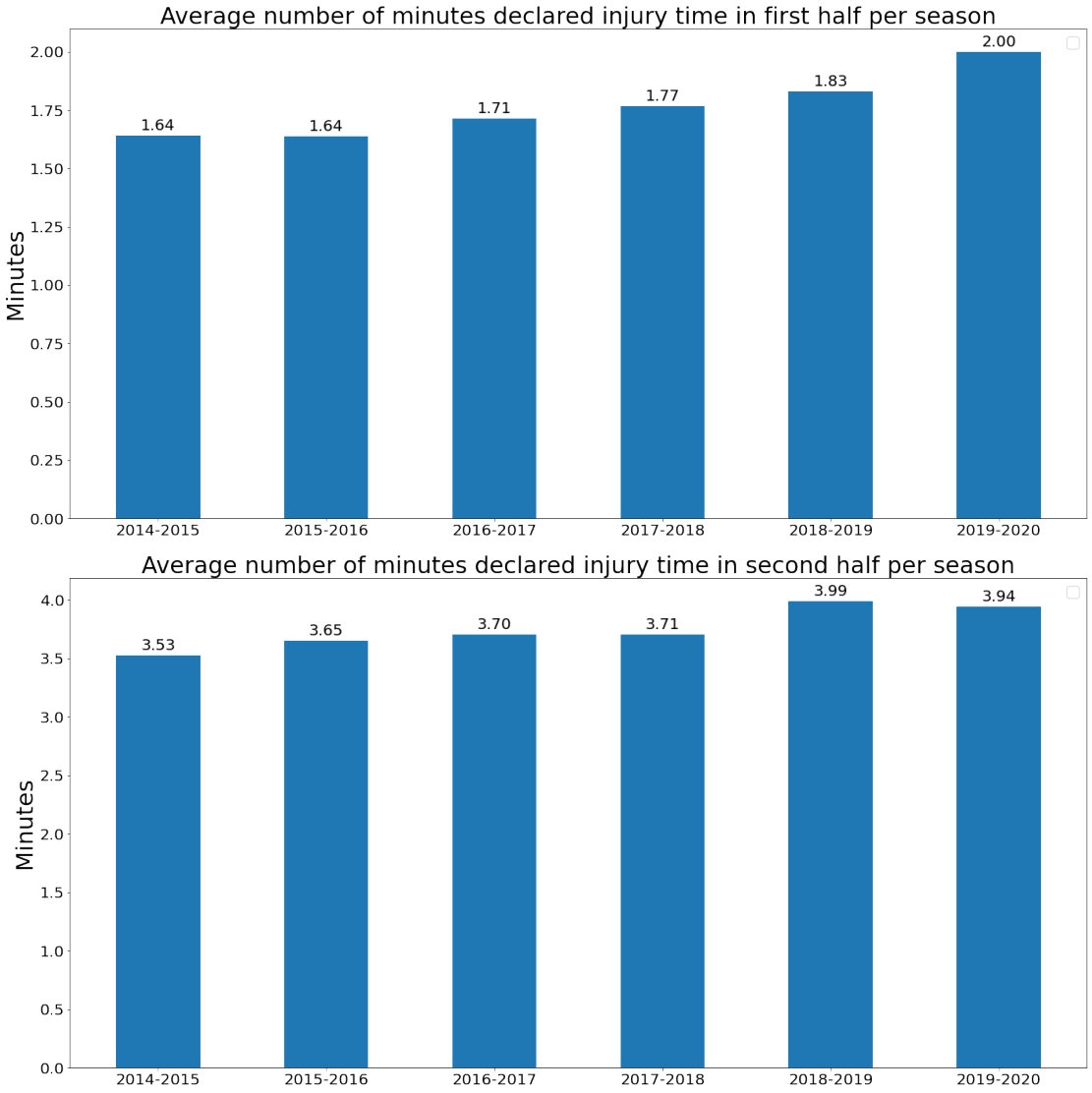


Figure 3.4: Average declared injury time in each season

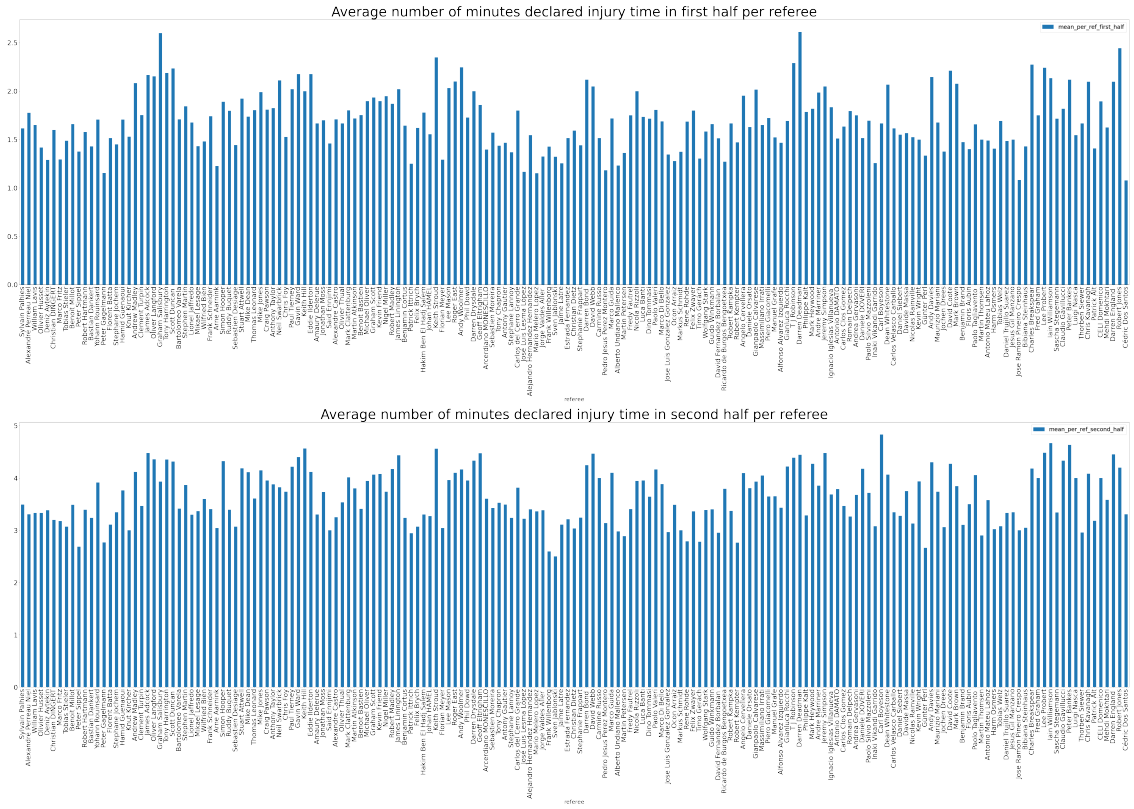


Figure 3.5: Average declared injury time for each referee, all referee's have more than 25 games in the dataset

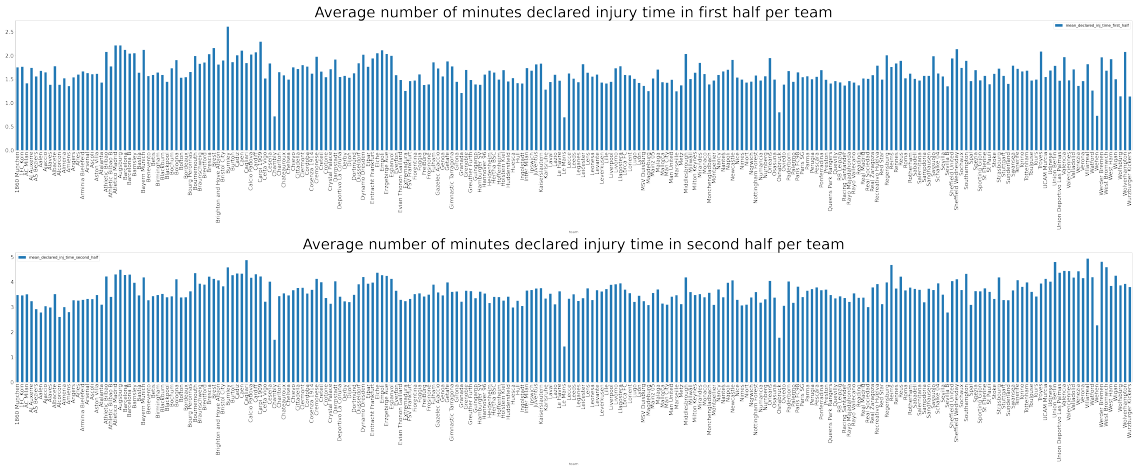


Figure 3.6: Average declared injury time for each team

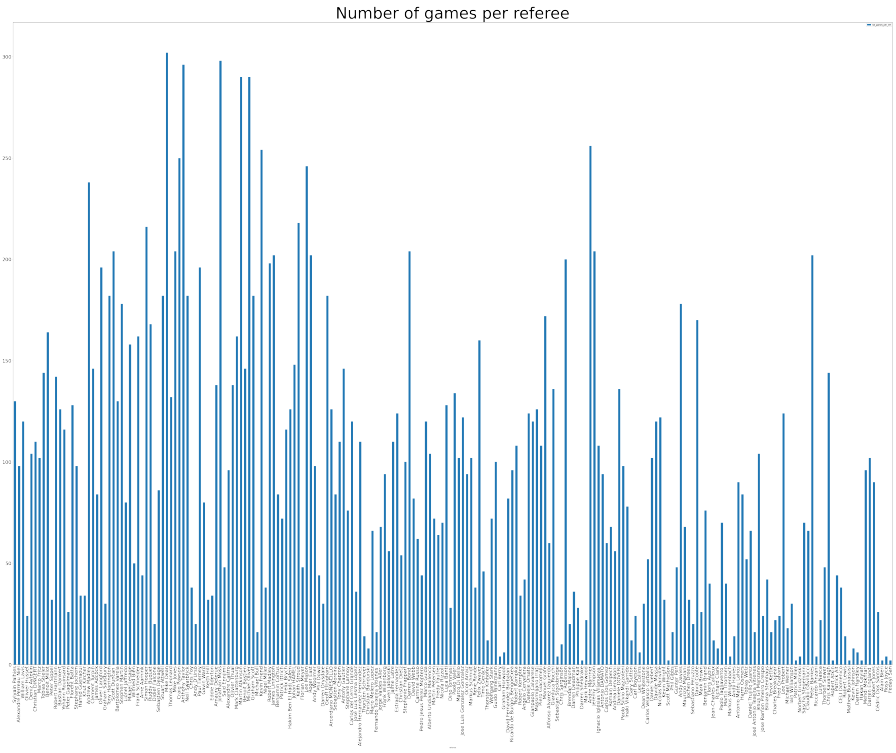


Figure 3.7: Number of games per referee

Chapter 4

Implementation

4.1 Google Colab

All the different models are built using Google Colaboratory, Colab. In Colab Python code is written in the browser and executed on Google cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine according to Google 2021.This is a good software for data scientist and machine learning programs. The code is written in blocks similar to Jupyter notebook and this gives a good overview of the code. Another advantage writing code in blocks is that not all code must be executed every time, only the selected blocks will be executed. Colab is integrated with Google drive, so all the notebooks, datasets and models are stored safely and easily accessible.

4.2 Pandas

"pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language" McKinney 2010. The pandas library is in Colab by default, hence there is no need for installation. Pandas is used to import data into a Python program. Usually data is saved in a .csv, .xlsx or a similar formatted file and pandas import and converts the data stored in such a file to a dataframe that can be easily be read and manipulated in a Python program. Dataframe is the primary pandas datastructure, it is two-dimensional, tabular and contains labeled axes. A big advantage with the dataframe structure is that is supports vectorized functions, which is much faster than using a traditional for-loop. Manipulating data is much easier and faster with vectorized functions.

4.3 Statsmodels

"statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration" Seabold and Perktold 2010. To build the some of the models, Linear model, Poisson model and Negative binomial model, statsmodels is an efficient tool. Using statsmodels, much of the code can be reused across the different models. A sublibrary to statsmodels is statsmodels.formula.api which is "a convenience interface for specifying models using formula strings and DataFrames. This API directly exposes the from_formula class method of models that support the formula API" Seabold and Perktold 2010. This includes ordinary least squared models, the linear regression model and generalized linear models such as the Poisson regression model and negative binomial regression model. The models are fitted using a formula, dataframe and statistical distribution. An example of the formula that was used for before game prediction is "declared_inj_time $\sim$ competition_name

+ referee + team1_name + team2_name".

4.4 Keras

"Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides" Gulli and Pal 2017. Keras is built on top of Tensorflow, one of the most used libraries within machine learning and the goal is to simplify code. The Keras library is used to build the neural network models, including the embedding layers. When building models using Keras, there are lots of different hyperparameters which can be tuned in order to receive optimal performance. In addition, Keras has a built in tuner to find the best hyperparameters for a model. Another advantage using Keras is the support of GPU, making it faster to train the model. It is easy to save and load models, especially when using Colab, since everything is saved and loaded directly on Google drive.

4.5 Matplotlib

"Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python" Hunter 2007. This library supports loads of different visualizations techniques. Matplotlib is a object-oriented API used to create plots in Python programs. It is built to be as usable as MATLAB, but with Python code. All plots, charts and data visualizations are created with Matplotlib.

4.6 Scikit-learn

Scikit-lean is a simple and efficient tool for machine learning in Python. In this thesis, Scikit-learn was used to preprocess the data, and specific the inputs to the ANN before the game. These inputs was encoded using the class label encoder from the scikit-learn library. The label encoder encodes target labels to values between 0 and n_classes-1 Pedregosa et al. 2011.

4.7 NumPy

NumPy is a numerical computing tool, it is fast and powerful, and easy to use. It includes array, vectorization and GPU support. In this thesis, NumPy is used to calculate errors, accuracy and the χ^2 statistic Harris et al. 2020.

4.8 SciPy

SciPy contains mathematical algorithms, statistics and convenience functions built upon NumPy for Python. It is used to calculate the p-value based on the χ^2 statistic from the predictions Virtanen et al. 2020.

Chapter 5

Results

In this section, the results of every model developed will be presented. The performances of the different models will be compared and tested to see which models give the most accurate measurements. All the models will be evaluated according to section 2.5. The dataset has split into a training set and a test set. The models have been trained on the training set and then used to predict injury times for games in the test set. It is the results from the predictions which will be evaluated. For each model, a confusion matrix and a distribution of prediction will be shown and discussed. The models output a floating number, and this prediction is used to calculate the point errors. Additionally, the floating number will be rounded to the closest integer and to calculate the χ^2 statistic and accuracy.

5.1 Before the game

In this section, a performance review will be conducted on the quality of the pregame models. The inputs are home team, away team, referee, and competition, and the output is a prediction of how many minutes will be added to the game. The models will be used to predict injury time in football games, and the performance of each of the models is presented accordingly to 2.5. Tables 5.1 and 5.2 shows MSE, RMSE, MAE, MAPE, accuracy, and the results from the χ^2 goodness of fit test, both χ^2 statistics and p-values, for all of the models. Additionally, the tables show that the second half models have better and more accurate predictions and overall a better fit than the first half models. When comparing the error measurements of the predictions, MSE is quite similar between the first and second half. However, this error is less significant in the second half models because, in the second half, there are higher numbers in the predictions and declared injury time, so MAPE is lower.

The MSE, RMSE, and MAE, describe the differences between the actual declared injury times and predicted injury time. The lowest difference between the actual declared values and predicted injury times is the predictions made by the NB model. From table 5.1, it is known that there is about one minute difference between actual injury times and predicted injury times, which corresponds to 48.11% error. MAE weights all the errors equivalently; meanwhile, RMSE weights big errors relatively higher, and the difference between MAE and RMSE is a measurement of variation in errors. If the difference is higher, the variance is higher in individual errors. In the first half, the outliers are high values of declared injury time, which the model will struggle to predict. Hence, if there is a smaller difference between MAE and RMSE, the model more often predicts more minutes whenever the actual declared injury time is high. The ANN model has the least difference. Hence, this model has fewer high errors. Regarding the accuracy of the models, the linear model predicts correctly 1% more than the others. All the models predict correctly about 40% of the time. The chi^2 statistic measures how good fit; the linear model has a value of 12005, while the ANN model has a fit of 394, meaning the fit of the ANN model is considerably better, even though the errors are a bit higher. The uncertainty on predictions from the models varies from 0.15 for the ANN

	Linear model	Poisson model	Negative binomial model	Regression ANN model
MSE	1.14	1.55	1.13	1.22
RMSE	1.07	1.25	1.06	1.11
MAE	0.76	0.87	0.75	0.82
MAPE (%)	48.11	49.34	48.10	55.47
Accuracy	0.41	0.40	0.41	0.40
χ^2 test (statistic, p-value)	12005.39, 0.0	721.53, 0.0	6318.50, 0.0	393.86, $4.7 * 10^{-85}$
Mean standard error of prediction	0.29	0.39	0.74	1.29
Mean coefficients home/away team	0.06/0.03	0.05/0.02	0.04/0.02	
Mean standard error on regressors	0.30/0.31	0.24/0.25	0.39/0.40	

Table 5.1: Performance by the models in first half

model to 0.27 for the Poisson model. Each regressor has its own estimated coefficient, the mean of all coefficients for the home team in the first half is found in table 5.1. To find the confidence interval for a regressors, the corresponding standard error is used. A measure of uncertainty in the regressors' home team and away team is the standard error. Hence the mean standard errors for all home teams and all away teams are presented in table 5.1. In the ANN model, each input does not have a corresponding coefficient. Table 5.1 also shows that all p-values are negligble values and all the models are rejected.

	Linear model	Poisson model	Negative binomial model	Regression ANN model
MSE	1.22	1.22	1.22	1.13
RMSE	1.11	1.11	1.11	1.07
MAE	0.82	0.82	0.82	0.80
MAPE (%)	26.04	26.02	26.02	23.54
Accuracy	0.39	0.40	0.40	0.41
χ^2 test (statistic, p-value)	1822.29, 0.0	954.83, 0.0	480.48, 0.0	2102.05, 0.0
Mean standard error of prediction	0.31	0.56	0.55	1.10
Mean coefficients home/away team	0.09/0.10	0.03/0.03	0.03/0.03	
Mean standard error on regressors	0.31/0.32	0.16/0.16	0.16/0.17	

Table 5.2: Performance by the models in second half

The linear model, Poisson model and NB model predicts has equal errors on their predictions. This is explained further in the sections to come. The ANN model has lower errors compared to the other models, and the mean error on each prediction is about one minute. The ANN model

has lower difference between RMSE and MAE. Hence, the ANN model has less high errors. MAPE shows that the percentage errors are lower compared to the first half, this can be explained by that the MSE, RMSE, and MAE are quite equal in both halves, however the injury times are in general higher in the second half. Hence, a one minute error in the second half will be a lower percentage of the declared injury time. The accuracy, is similar to the accuracy in the first half, the models predicts correctly about 40% of the time. All the χ^2 statistics are higher compared to the ANN model in the first half, and similar to the first half, all the p-values are zero. Hence the models are rejected. In the first half, the ANN model has highest errors, and lowest χ^2 statistic, however this is opposite in the second half. In the second half, the ANN model has the lowest errors, and highest χ^2 statistic. In general, it seems all of the models, both in first and second half, miss on its predictions with about a minute.The mean standard error is the uncertainty in the model, and prediction interval. The linear model has the lowest uncertainty on its predictions, however the regressors have higher uncertainties. All of the p-values are lower than 0.05, and all of the models are rejected.

5.1.1 Linear model

A null hypothesis for the linear model is stated: The declared injury time can be predicted by a fitted linear model without a significant error.

First half

Figure 5.1 shows the distribution of rounded predictions. In most of the games the model predicts two minutes of injury time, in some of the games one minute, and in a few games it predicts three or four minutes. From figure 5.2, the number of correct predictions and wrong predictions are known. The model has highest accuracy, when the predicting one minute, and the accuracy is 63.01%, and overall the model predicts correctly in 41% of the games. For a linear model, a measure of goodness of fit is R^2, described in section 2.1.3. For this model $R^2 = 0.132$. Regarding R^2, there are a small to almost none correlation between the declared injury time and the linear model. It is likely that declared injury time can not be described by a linear relationship between the inputs in the first half. The results from the χ^2 goodness of fit test is to reject the null hypothesis. Hence, this model is not a good model for predicting injury time in a football match and the model is rejected.

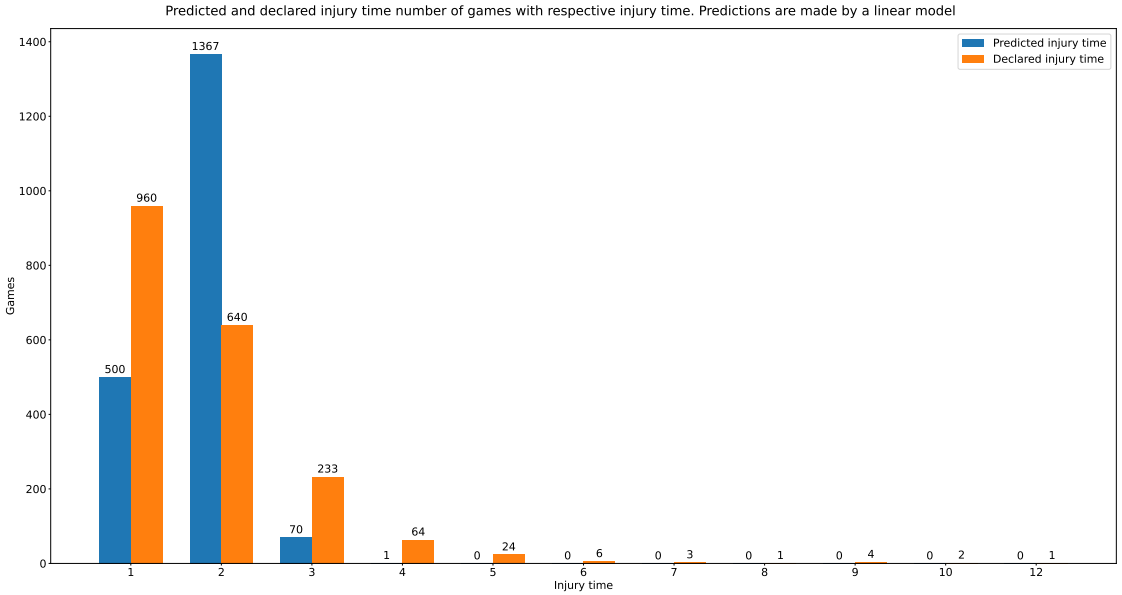


Figure 5.1: Counts of predicted injury times and declared injury times in first half from linear model

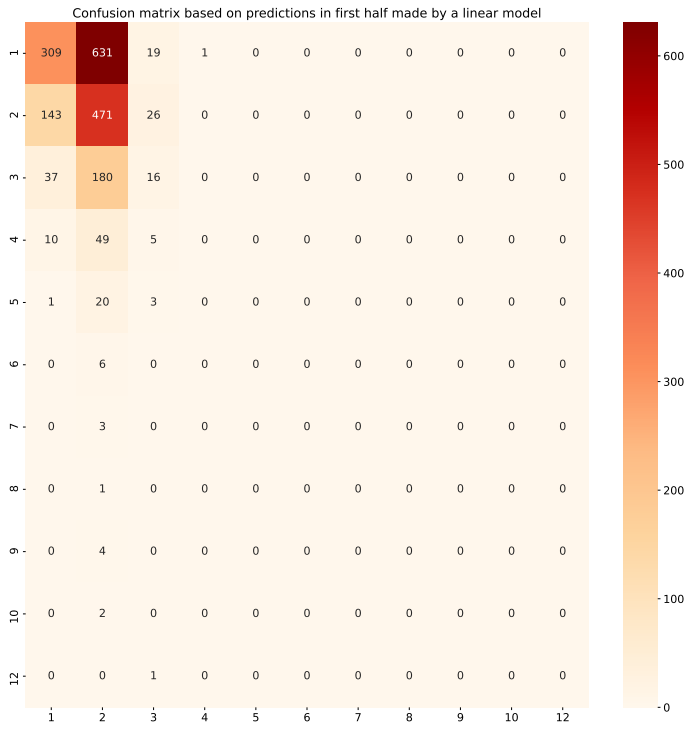


Figure 5.2: Confusion matrix of predicted injury times and declared injury times in first half from linear model

Second half

Figure 5.3 shows the distribution of predictions compared to the actual injury time distribution.When inspecting Figure 5.4, the confusion matrix shows that the linear model has most accurate predictions when predicting four minutes of injury time, and the accuracy is 43.74%. Overall the model predicts correctly about 40% of the time, meaning the model does not perform significantly better on any specific prediction. The calculated $R^2 = 0.237$, and it is unlikely that declared injury time can be described by a linear relationship between the inputs in the second half. The χ^2 goodness of fit test results suggest the same, with a p-value of zero. Hence, the model is not a good fit and the model is rejected.

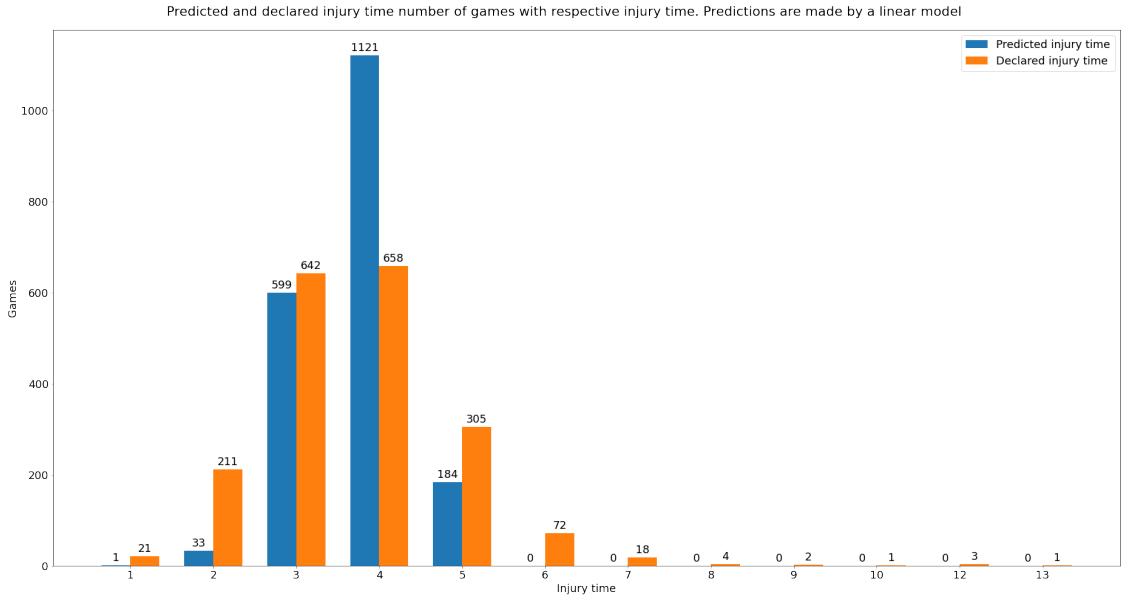


Figure 5.3: Counts of predicted injury times and declared injury times in second half from linear model

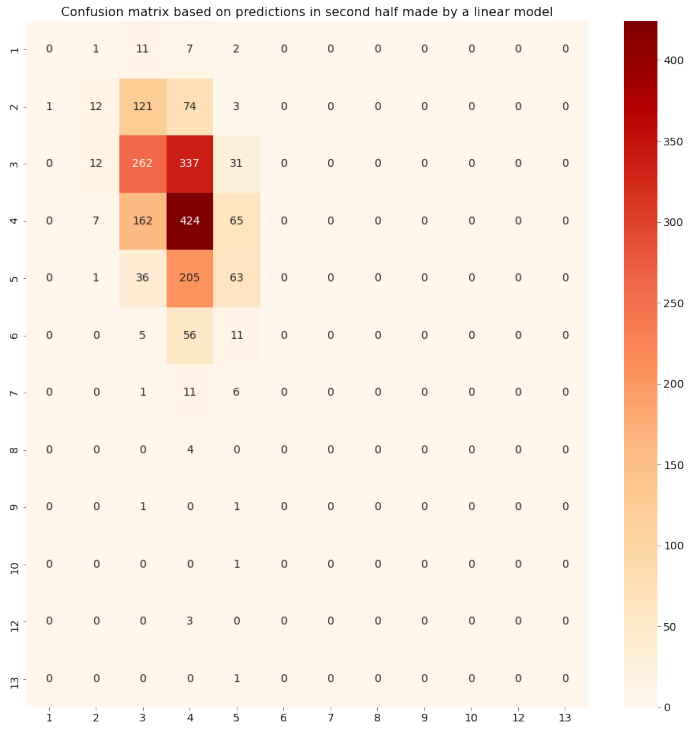


Figure 5.4: Confusion matrix of predicted injury times and declared injury times in second half from linear model

5.1.2 Poisson model

When using a Poisson model and a χ^2 goodness of fit test on the results the null hypothesis is: The declared injury time can be predicted by a Poisson model without a significant error. The results will either reject or fail to reject this null hypothesis.

First half

Figure 5.5 shows the overall expected counts compared to the actual declared injury times. The big difference is games with no declared injury time, these data are corrupt in the dataset. Hence, there are no recorded games with zero minutes of injury time, however, a Poisson model expects zero a certain amount of times. Underdispersion occurs, and the actual counts have less spread compared to the expected counts. Figure 5.6 shows the confusion matrix for rounded values of the mean. The model is most accurate when predicting one minute of injury time, with an accuracy of 48.08%. Overall, the model predicts correctly 40% of the time, found in table 5.2. A χ^2 goodness of fit test was conducted and the results gives a p-value of 0.0, hence the null hypothesis is rejected. There is significant difference between the predicted injury time and the actual declared injury time and the model is rejected.

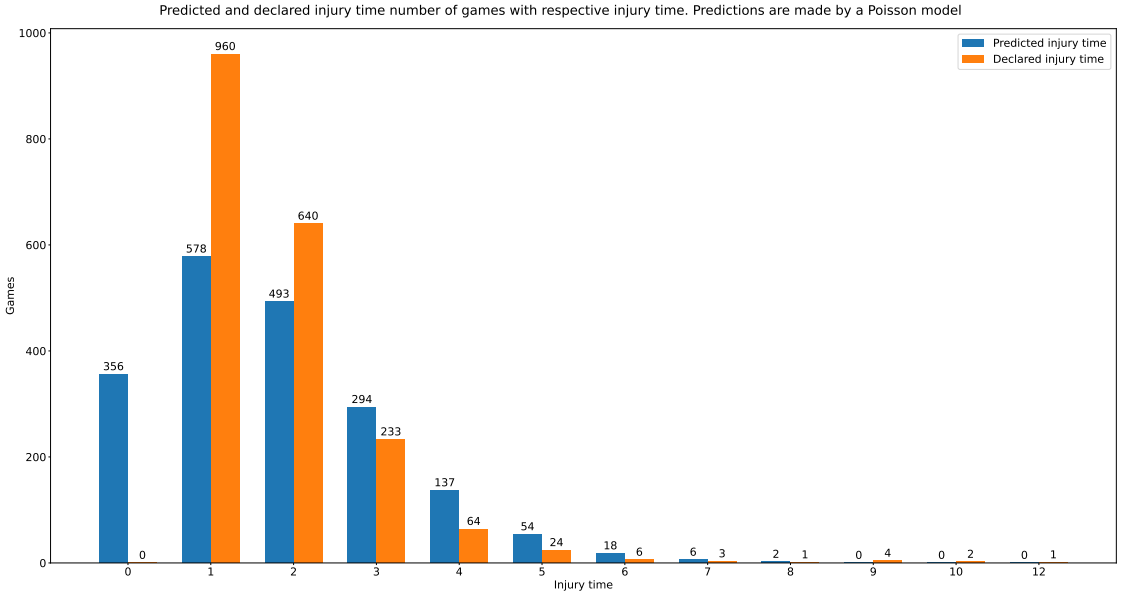


Figure 5.5: Counts of predicted injury times and declared injury times in first half from Poisson model

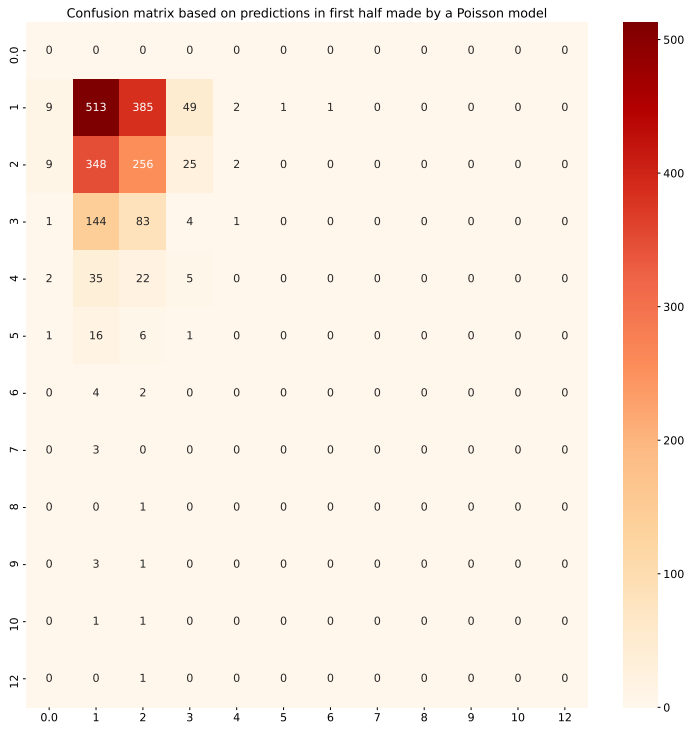


Figure 5.6: Confusion matrix of predicted injury times and declared injury times in first half from Poisson model

Second half

Figure 5.7 shows the expected frequencies by the model compared to the actual declared injury time frequencies. Underdispersion is present, and the actual counts lower spread compared to the expected counts. Figure 5.8 shows the confusion matrix comparing the actual injury time and rounded predicted injury time. When the model predicts three minutes of injury time, it is the most accurate prediction and it is correct 43.85% of the time. However, the overall accuracy is about 40%, meaning the model is not especially better at predicting any specific minute. A χ^2 goodness of fit test was conducted and the results shows that the null hypothesis should be rejected and that there is a significant difference between the predicted values from the model and the actual declared injury minutes. Hence, the model is rejected.

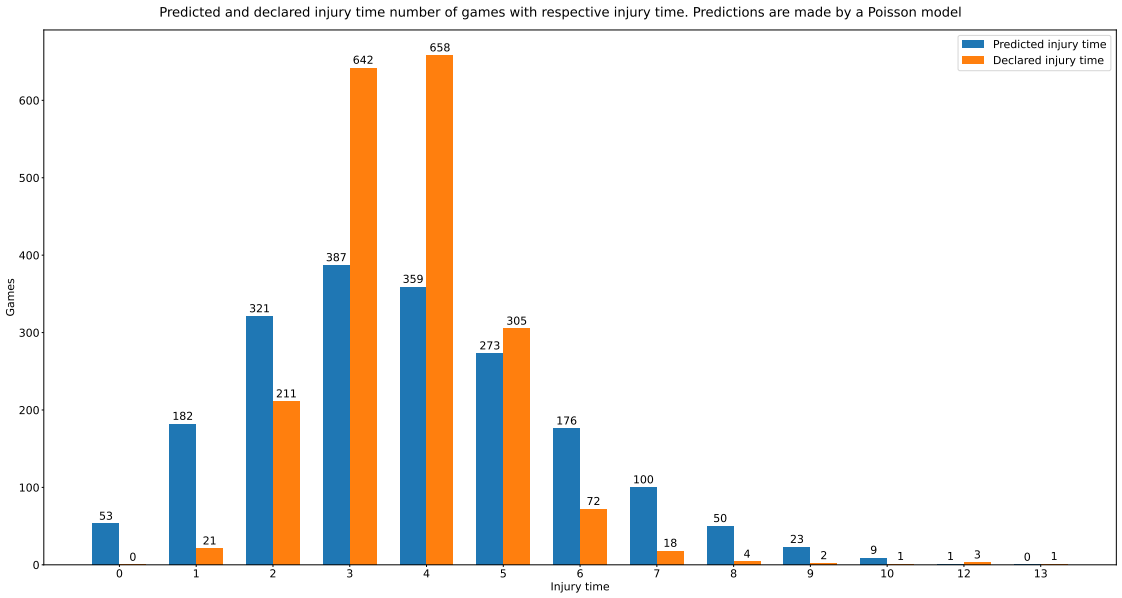


Figure 5.7: Counts of predicted injury times and declared injury times in second half from Poisson model

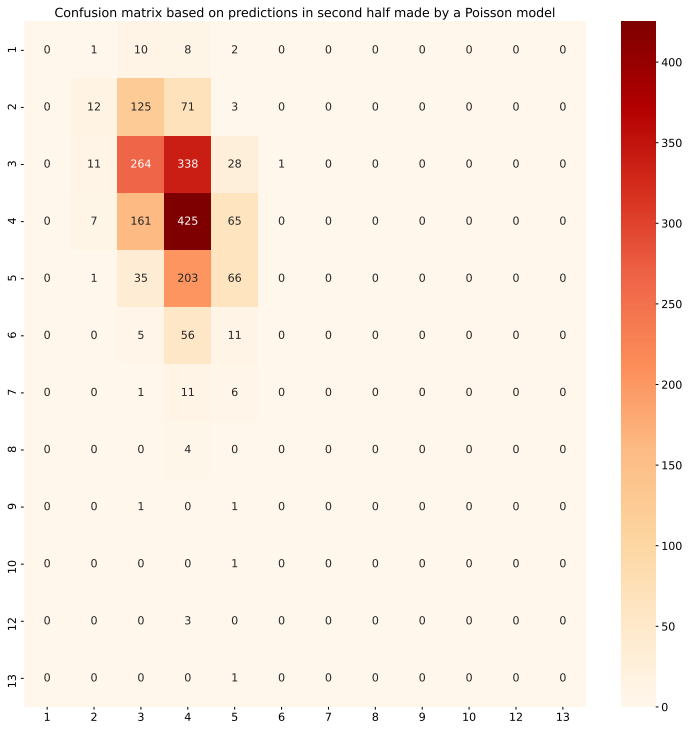


Figure 5.8: Confusion matrix of predicted injury times and declared injury times in second half from Poisson model

5.1.3 Negative binomial model

The null hypothesis for the χ^2 goodness of fit test completed on the predictions from this model is: The declared injury time can be predicted by a negative binomial model without a significant error. The results will either reject or fail to reject this null hypothesis.

First half

Figure 5.9 shows the expected frequencies from the NB model compared to the actual declared frequencies of injury time. The NB model provides a better fit to the underdispered data, and almost successfully removes zero. Compared to the Poisson model, this model more often predicts higher values, and values higher than three minutes in the first half is unlikely. Figure 5.10 shows the confusion matrix based on rounded predictions. Based on this, the model predicts most accurately when predicting one minute with an accuracy of 62.16%. Compared to the overall accuracy found in table 5.1, which is 41%, the model has much better predictions when predicting one minute. The results from the χ^2 goodness of fit test rejects the null hypothesis. Hence there is a significant difference between the fitted negative binomial model's predictions and the declared injury times, and the model is rejected.

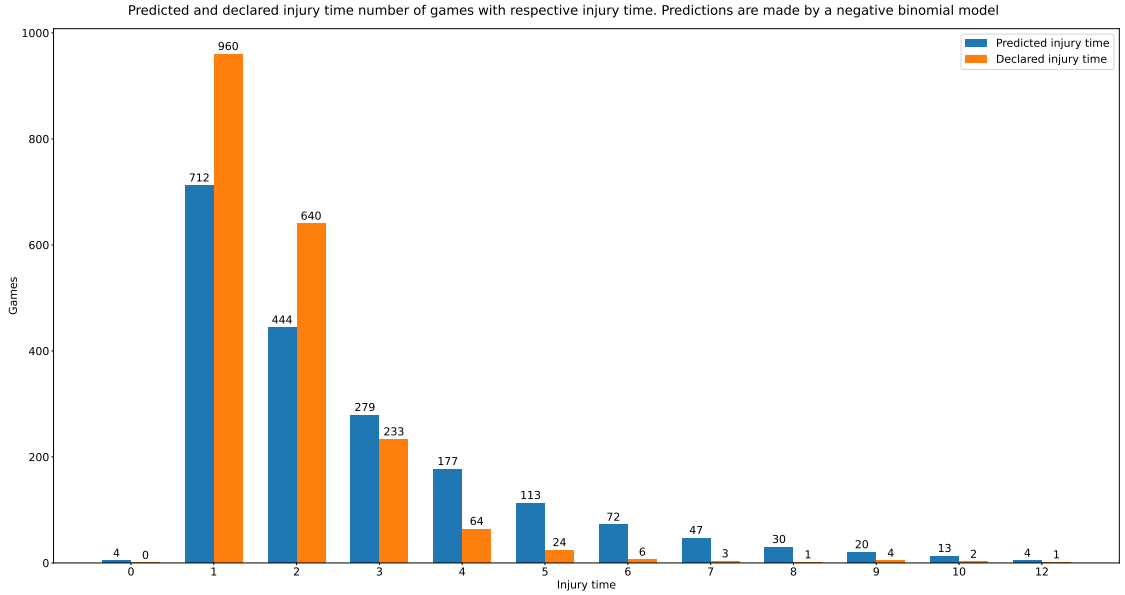


Figure 5.9: Counts of predicted injury times and declared injury times in first half from negative binomial model

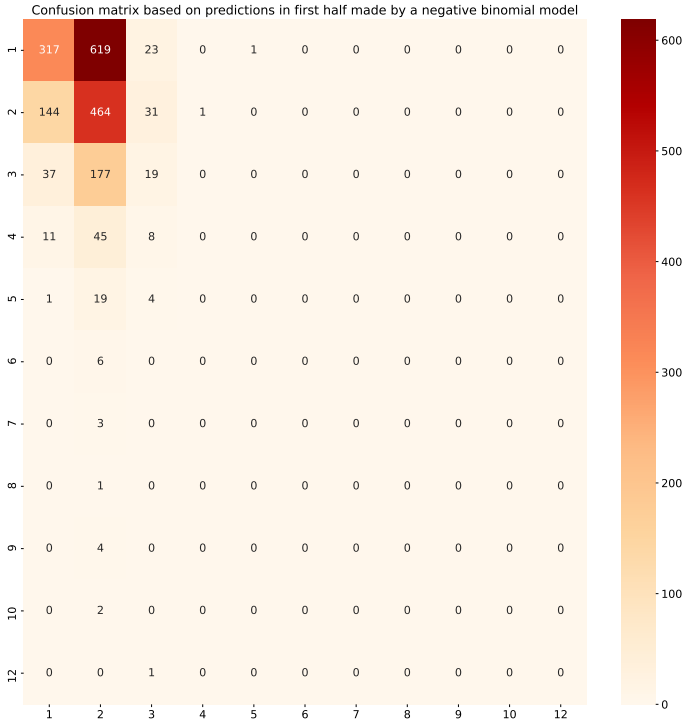


Figure 5.10: Confusion matrix of predicted injury times and declared injury times in first half from negative binomial model

Second half

Figure 5.11 shows the expected frequencies compared to the actual declared injury times. Compared to the Poisson second half model, this model has lower spread, but not as low as the actual declared injury times. When comparing first and second half, by looking at expected counts, the first half model look like a better fit. From figure 5.12, the confusion matrix based on rounded values is known, and the model is most accurate when predicting three minutes with an accuracy of 49.97%. Comparing this with the overall accuracy, found in table 5.2, the model predicts correctly about 40% of the time. A prediction of three minute is more accurate than other predictions. A χ^2 goodness of fit test was conducted, and the statistic is lower compared to the other models, suggesting a better fit. However, the p-value is less than 0.05 and there is a significant difference between actual declared injury times and predicted injury times. Hence, the model is rejected.

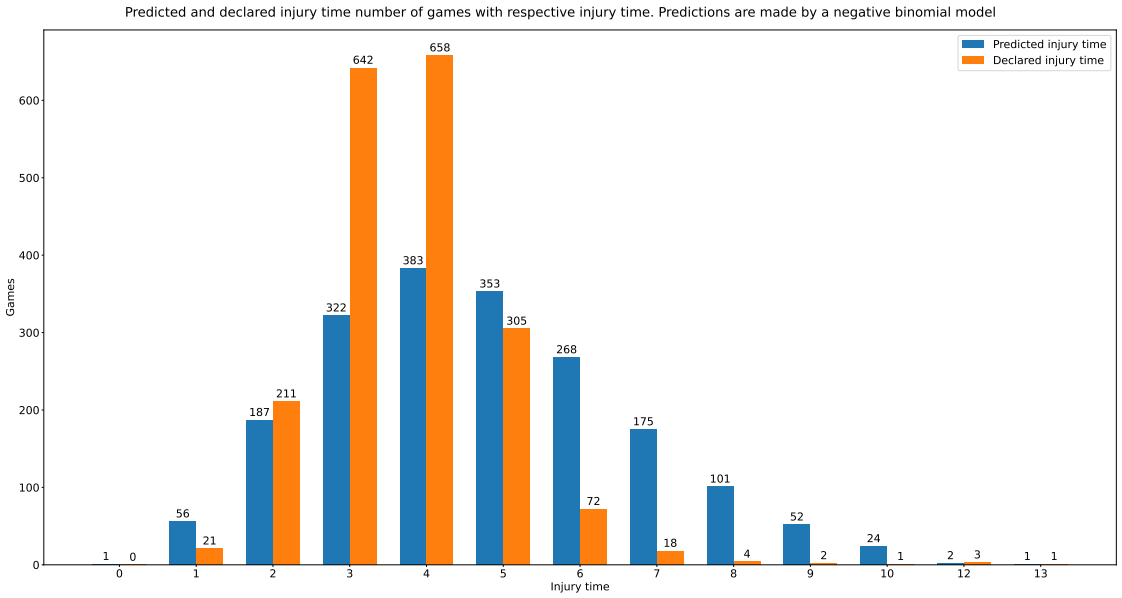


Figure 5.11: Counts of predicted injury times and declared injury times in second half from negative binomial model

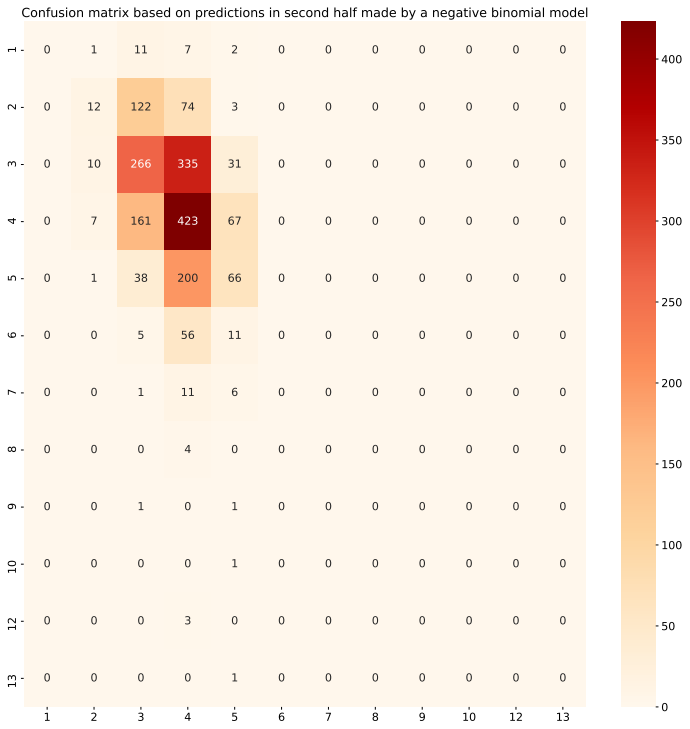


Figure 5.12: Confusion matrix of predicted injury times and declared injury times in second half from negative binomial model

5.1.4 Regression artificial neural network

A χ^2 goodness of fit test has been completed with the null hypothesis: The declared injury time can be predicted by a regression ANN model without a significant error. The results will either reject or fail to reject this null hypothesis. Both the ANN models has four input layers and four embedding layers. The inputs are home team, away team, referee and league. First step of the network is to transform the string inputs into vectors using an embedding layer for each of the input layers. The embedding layers are trained during the training process of the rest of the network.

First half

Figure 5.15 shows the resulting neural network after using Keras BO hyperparameter tuner. The ANN model consists of four inputs, four embeddding layers, six hidden layers with 104, 88, 88, 40, 8, and 8 neurons respectively and one output layer. After tuning, tanh was giving best results as an activation function and the stochastic gradient descent yielded best performance for the optimizer. The model was trained for 150 epochs, and MSE on the validation set was saved for every epoch. The model was then retrained to the epoch which yielded lowest MSE, which was 62.

Figure 5.13 shows the distribution of rounded predictions compared to actual declared injury times. The model most often predicts two minutes, one part of the reason is because the mean declared injury time is 1 minute and 43 seconds. Figure 5.13 shows the confusion matrix, and the model has the most accurate predictions when predicting one minute of injury time, with an accuracy of 55.76%. Compared to the overall accuracy, found in table 5.1, of 40%, the model performs better when predicting one minute. The results from the conducted chi^2 goodness of fit test, are to reject the null hypothesis. Hence there is a significant difference between the predicted injury times and the declared injury times, and the model is rejected.

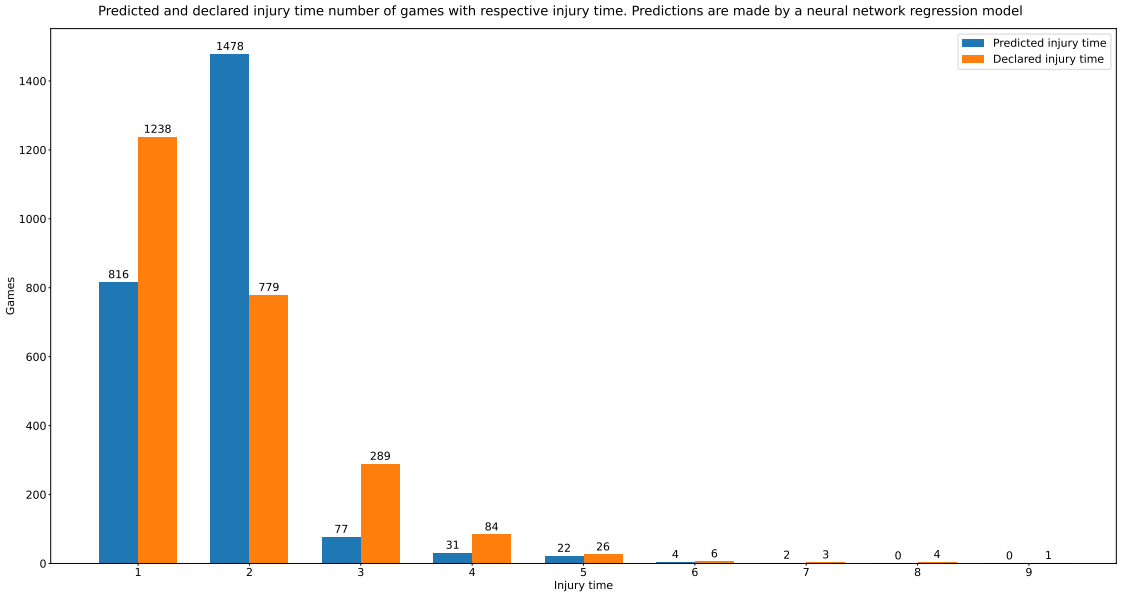


Figure 5.13: Counts of predicted injury times and declared injury times in first half from regression neural network model

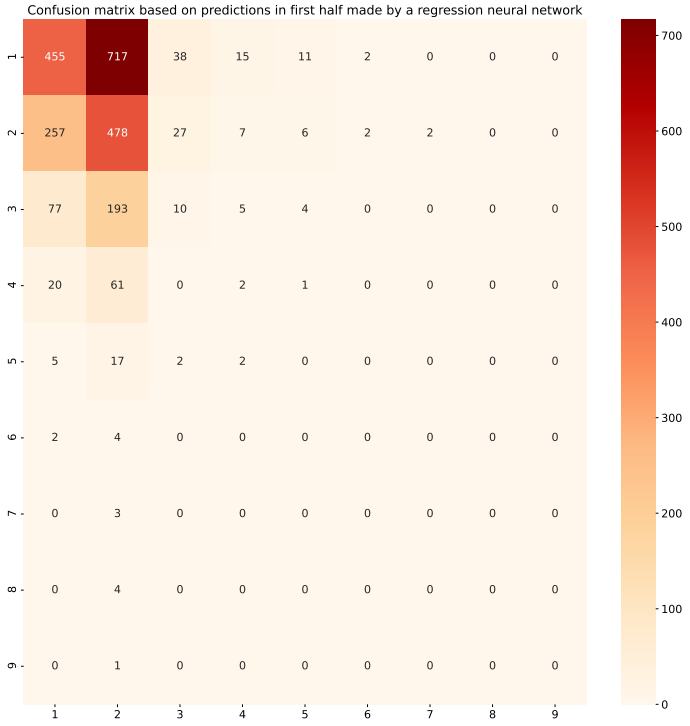


Figure 5.14: Confusion matrix of predicted injury times and declared injury times in first half from regression neural network model

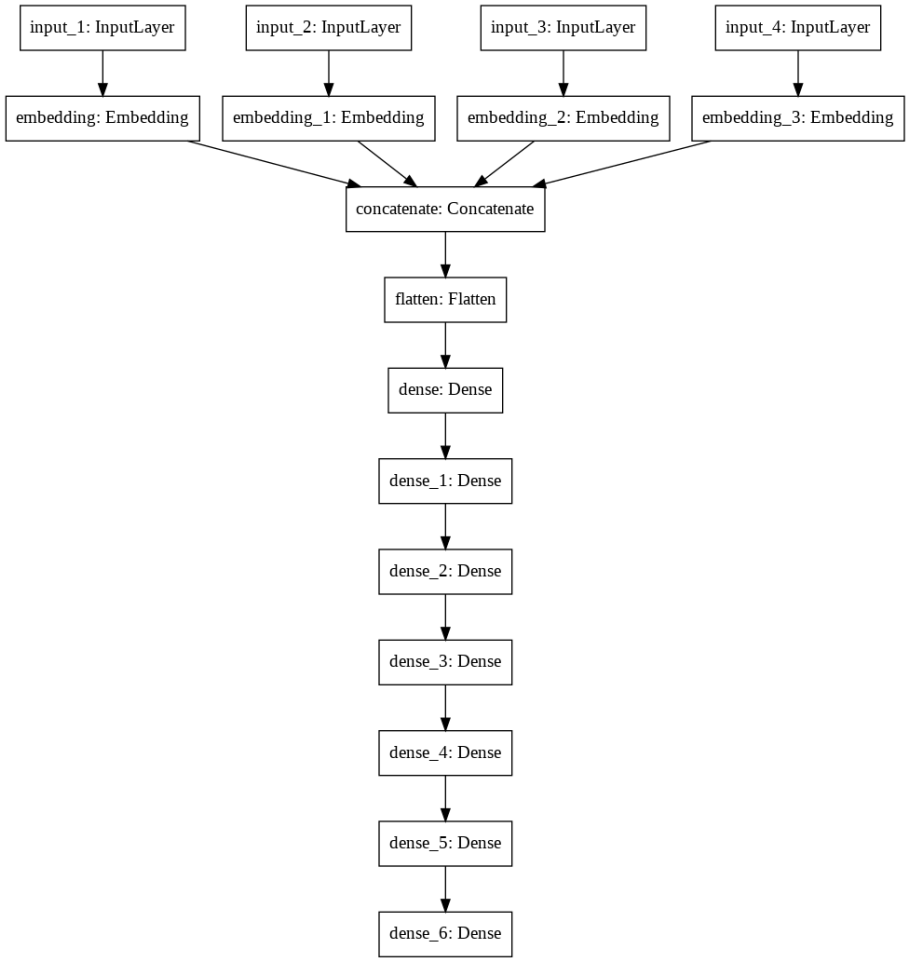


Figure 5.15: Resulting model after tuning hyperparameter using Keras tuner

Second half

Figure 5.18 shows the resulting neural network after using Keras hyperparameter tuner with Bayesian optimization. The ANN model consists of four inputs, four embeddding layers, three hidden layers with 40, 8, 8, neurons respectively and one output layer. After tuning, the linear function performed best as an activation function and the stochastic gradient descent yielded best performance for the optimizer. The model was trained for 150 epochs, and MSE on the validation set was saved for every epoch. The model was then retrained to the epoch which yielded lowest MSE, which was 7.

Figure 5.16 shows the frequencies of rounded predicted injury times compared to actual declared injury time frequencies. The model predicts most often four minutes of injury time, due to the mean being 3 minutes and 44 seconds. Figure 5.17 shows the confusion matrix, and the model is most accurate when predicting three minutes with an accuracy of 43.68%. Compared to the overall accuracy of about 41% the model predicts consistently for all minutes. From the χ^2 goodness of fit test, the results are to reject the null hypothesis. Hence there is a significant difference between the predicted values and the actual values and the model is rejected.

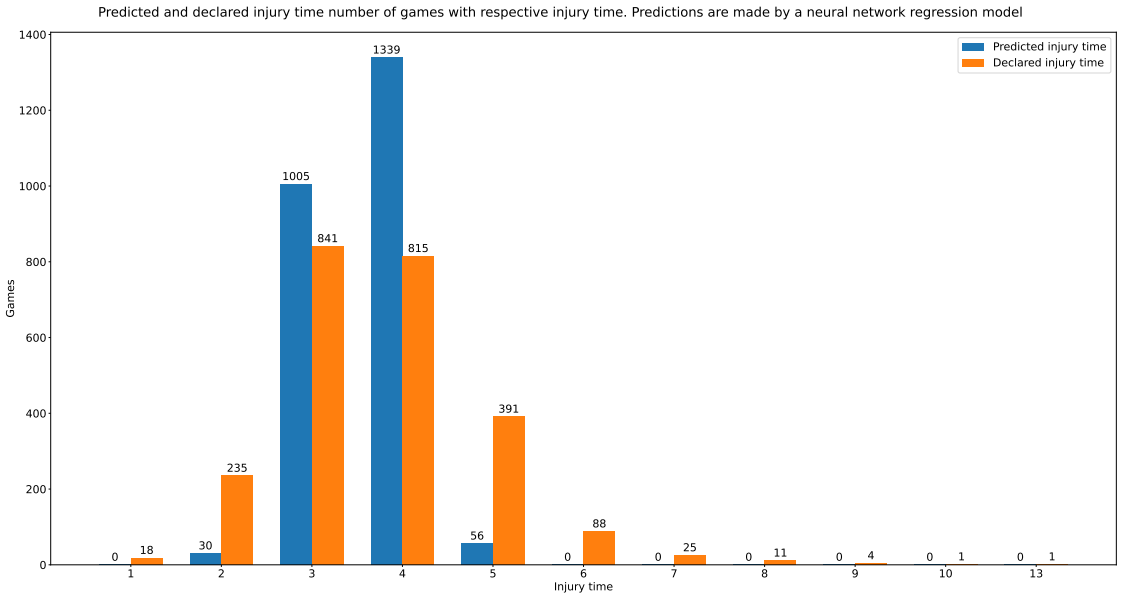


Figure 5.16: Counts of predicted injury times and declared injury times in second half from regression neural network model

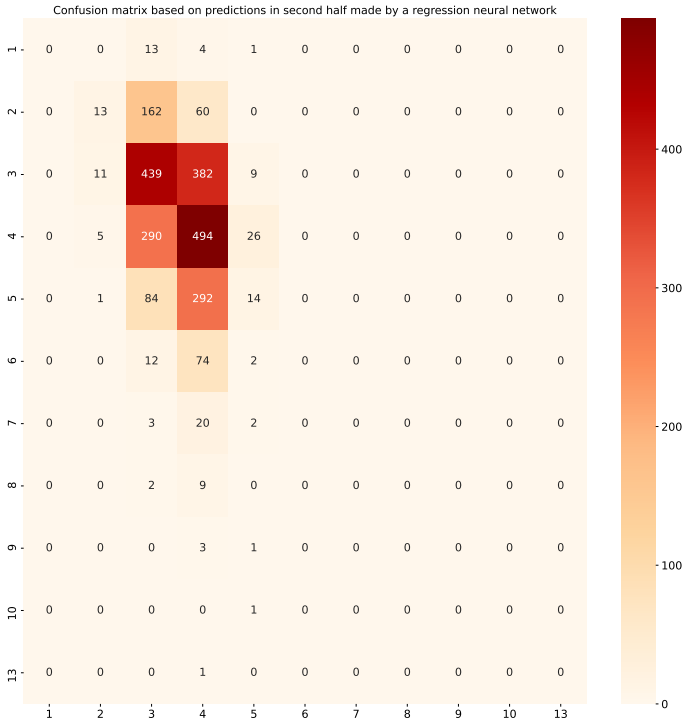


Figure 5.17: Confusion matrix of predicted injury times and declared injury times in second half from regression neural network model

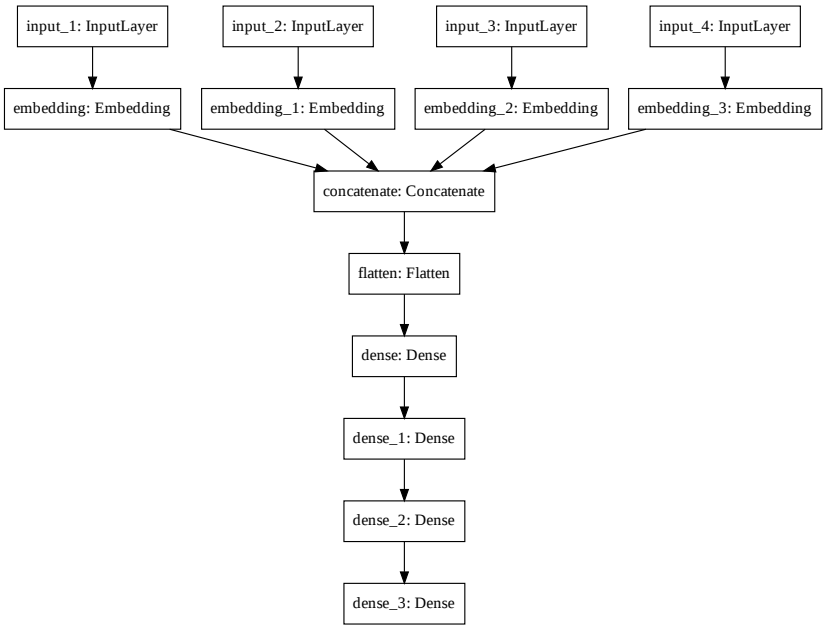


Figure 5.18: Resulting model after tuning hyperparameter using Keras tuner

5.2 Real-time prediction

In this section the models was built to predict injury time real time. The models is fed data from what happened in the game every five minutes and the inputs to each of the models are: pregame prediction (using the same pregame model), time of the data, number of goals so far in the game, number of substitutions so far in the game and how many seconds the game has been delayed. The performance for each model will be evaluated at every time step using equal measures as with the before game models, meaning the MSE, RMSE, MAE, MAPE, accuracy, and χ^2 goodness of fit test will be plotted at every time step for every model. For every model, a null hypothesis will be stated and the results from the χ^2 test will either reject the null hypothesis or fail to reject the null hypothesis at every timestep. As with the pregame models, the predictions are rounded to the nearest whole number. The live predictions are first made after the first time step at right before five minutes have passed and make new predictions every five minutes until the 94th minute, one time step after full time.

5.2.1 Linear model

The null hypothesis for this model is equal to the null hypothesis for the pregame linear model: The declared injury time can be predicted by a fitted linear model without a significant error.

First half

Figure 5.19 shows the error metrics, accuracy and the χ^2 goodness of fit test results with time on the x-axis. The error metrics, MSE, RMSE, MAE, and MAPE are lower at all time steps compared to the pregame predictions. Regarding the accuracy, at the beginning of the game the model has the same amount of correct predictions as before the game, but this increases all the way until the end of the half. At the end of the half, the model predicts correctly about 55% of the time. From figures 5.20 and 5.21 the distributions of predictions and confusion matrices are known at every timestep. The distributions at the start of the game looks similar to the pregame prediction

distribution shown in figure 5.1, however during the game the distributions decreases two minute predictions and increase one and three minute predictions, hence the at the end of the half, the predicted injury times looks more similar to the actual injury times. From the confusion matrices it is known that the amount of correct one and three minute predictions increase during the half and the amount of correct two minute predictions decrease. Even though the amount of correct two minute prediction decreases, the percentage correct predictions increase. At the end of the half (44:59), the model has the highest accuracy when predicting one minute of injury time and the accuracy is 79.64%. The χ^2 goodness of fit result shows that the fit increases, however the p-value is zero at all times. Hence, the model is rejected.

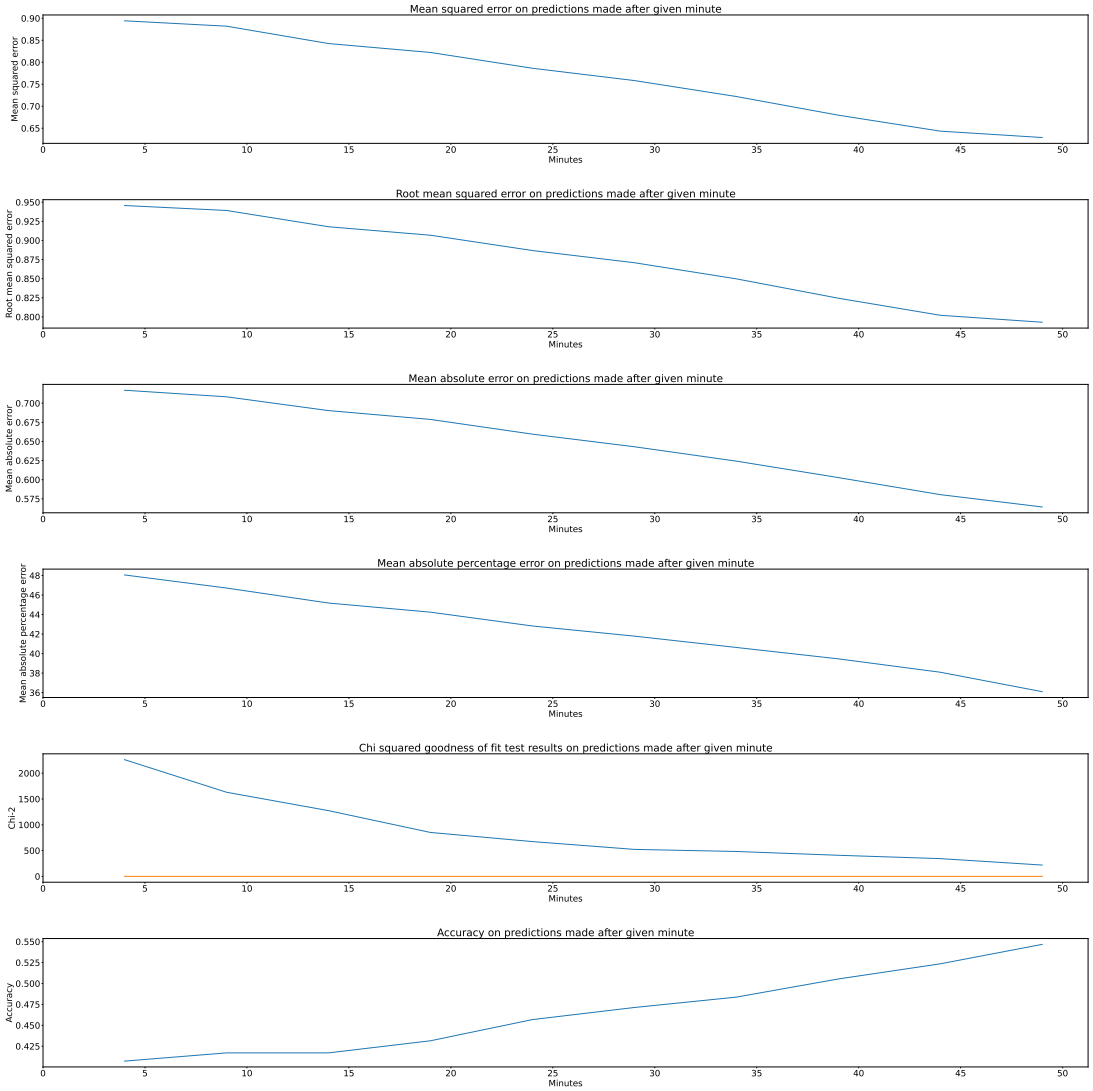


Figure 5.19: Performance on live predictions made by a linear model

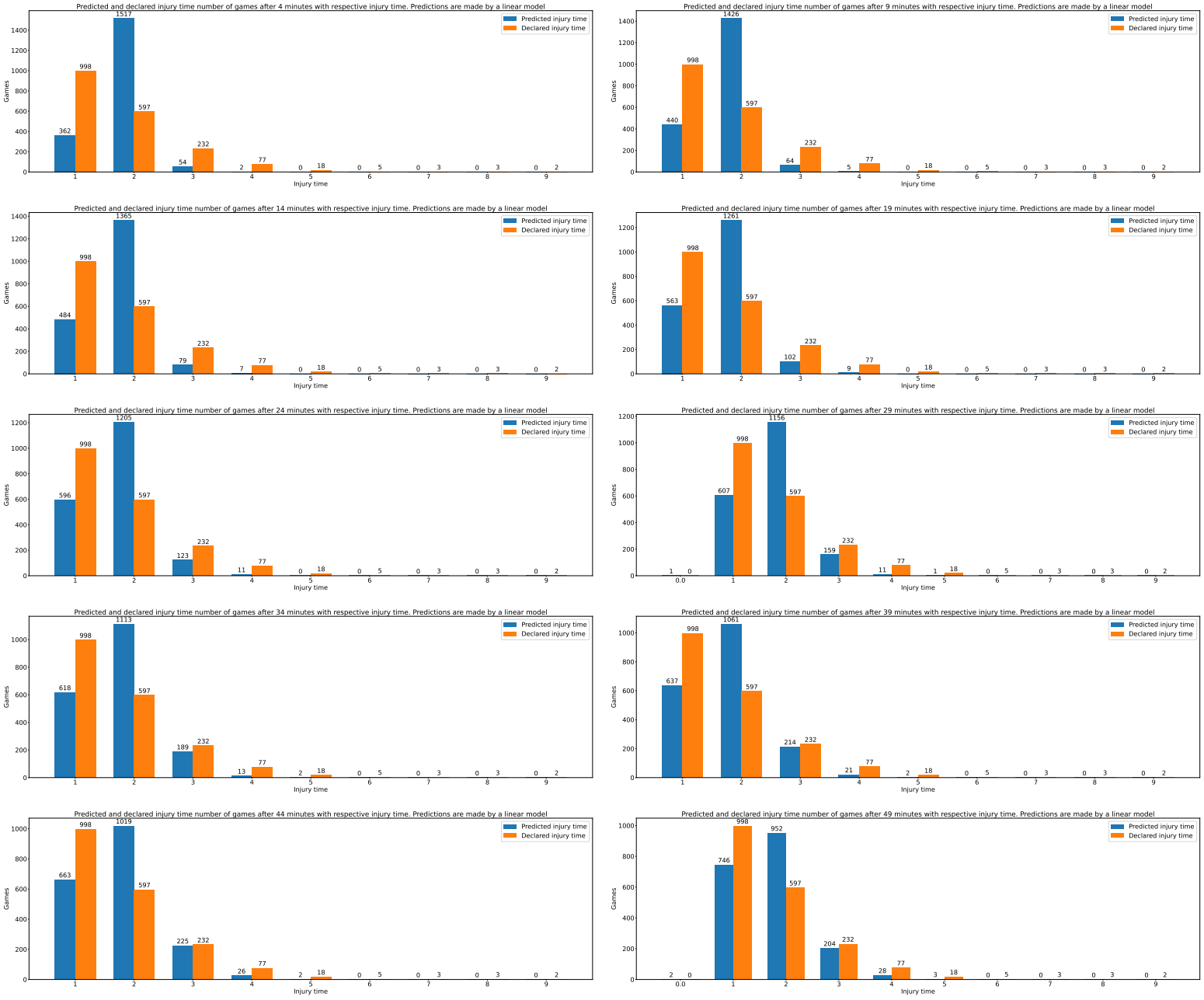


Figure 5.20: Distributions of predicted injury time and declared injury time at every time step

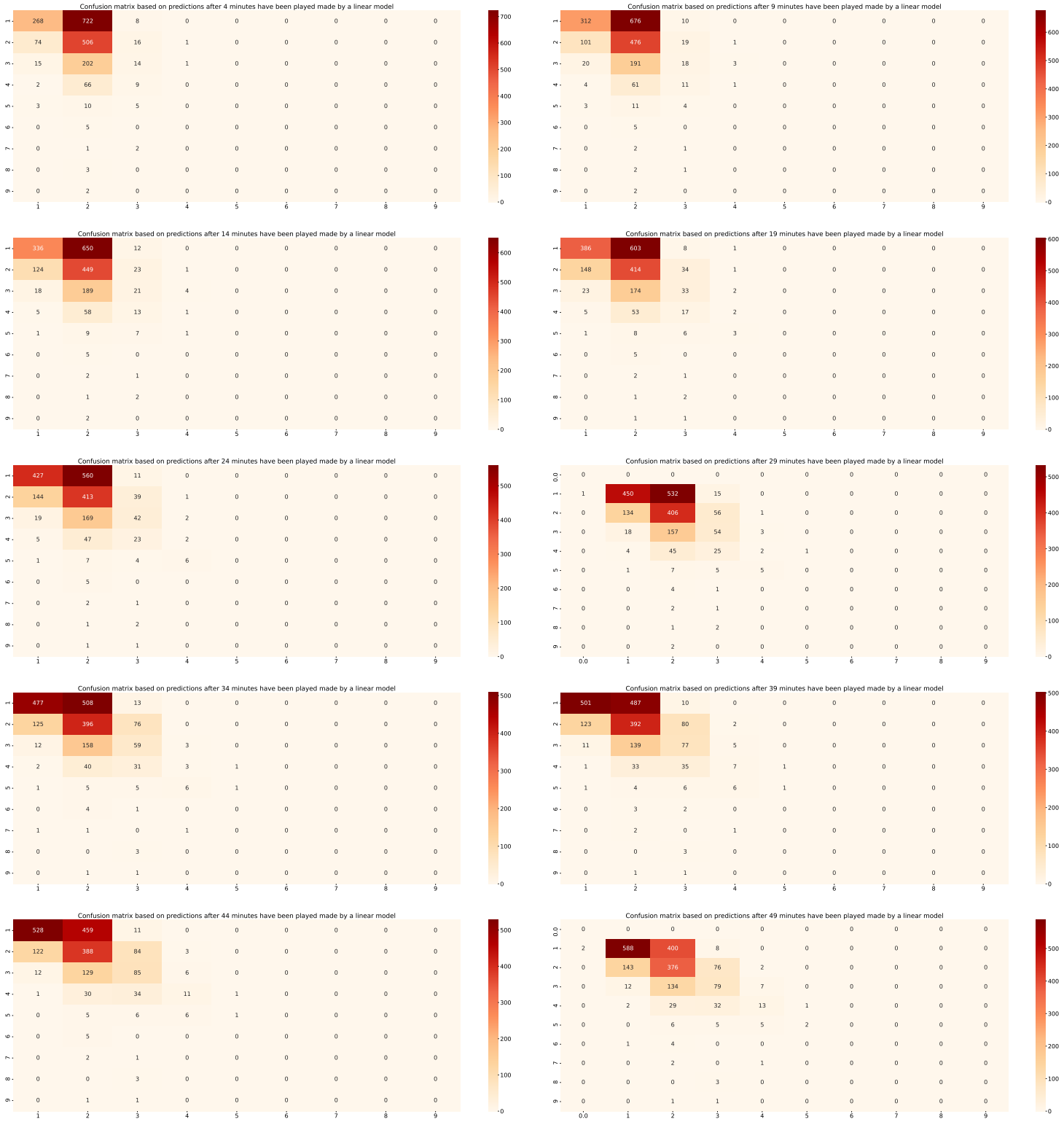


Figure 5.21: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

Second half

Figure 5.22 shows the MSE, RMSE, MAE, MAPE, accuracy and results of the χ^2 goodness of fit test for the linear model for live predictions for all timesteps with time on the x-axis. The errors are at all timesteps compared to the before game model. Like the first half model the errors decrease when time passes in a game. In addition, the χ^2 test results decrease and accuracy increases until almost 0.50%. Figures 5.23 and 5.24 shows the distribution of predicted injury times and actual injury times, and the confusion matrices at the different time steps. At the beginning of the second half, most of the predictions are four minutes, this amount decreases as time passes. On the other hand, the amount of two, three, five and six minute predictions increases. The confusion matrices shows that the amount of correct predictions follows a similar patterns at the distributions, correct two, three, five and six minute predictions increases, and correct four minute predictions decreases. The four minute accuracy increases, meaning that even though the model has less correct predictions, it predicts correctly a higher amount of the time. At the end of the game (89:59), the model has highest accuracy when predicting three minutes with an accuracy of 51.97%. From the χ^2 test results, it can be seen that the fit of the model improves, but there is still a significant difference between actual injury time and predicted injury time, due to the p-value being zero for at all times. Hence, the model is rejected.

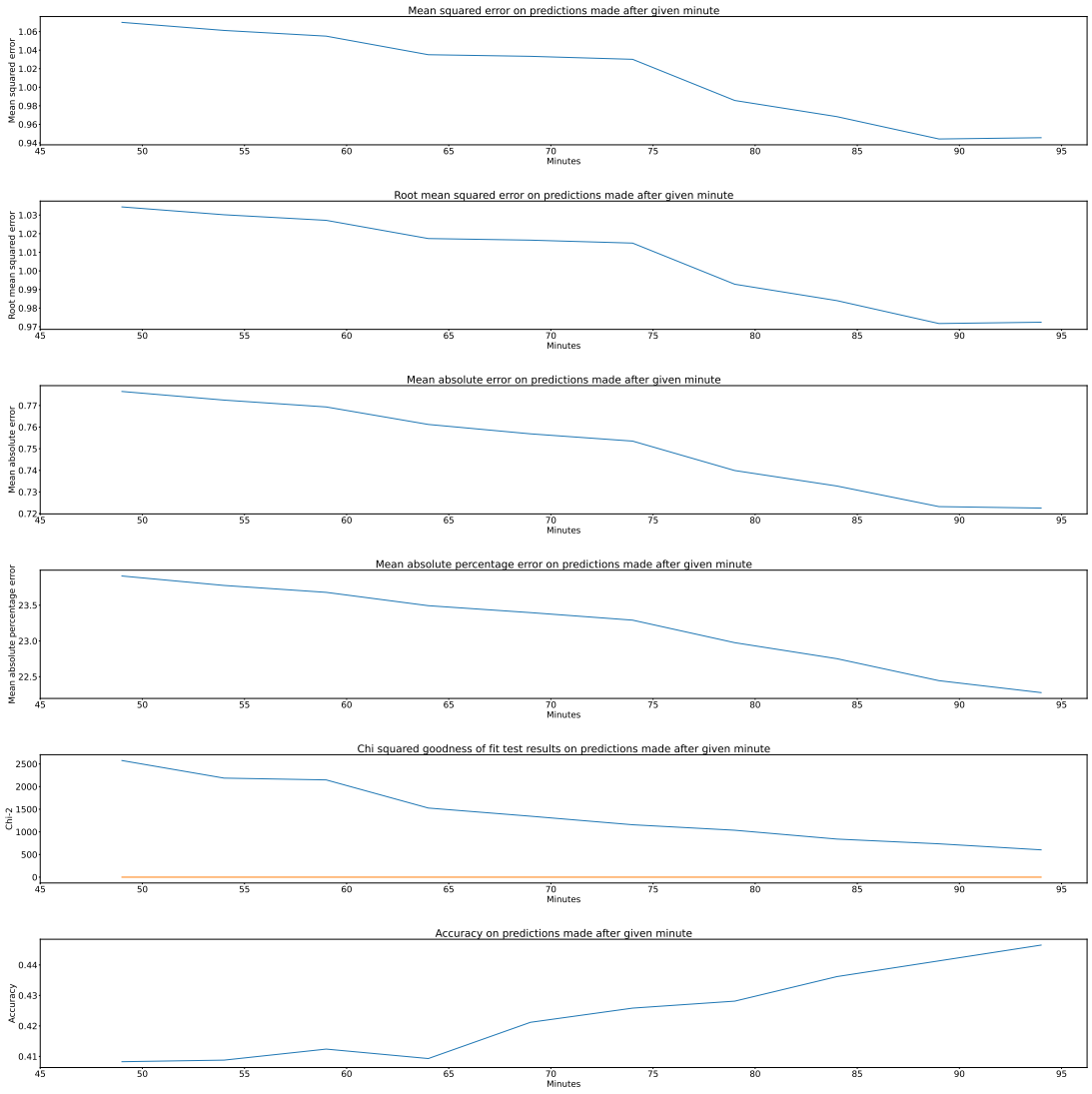


Figure 5.22: Performance on live predictions made by a linear model

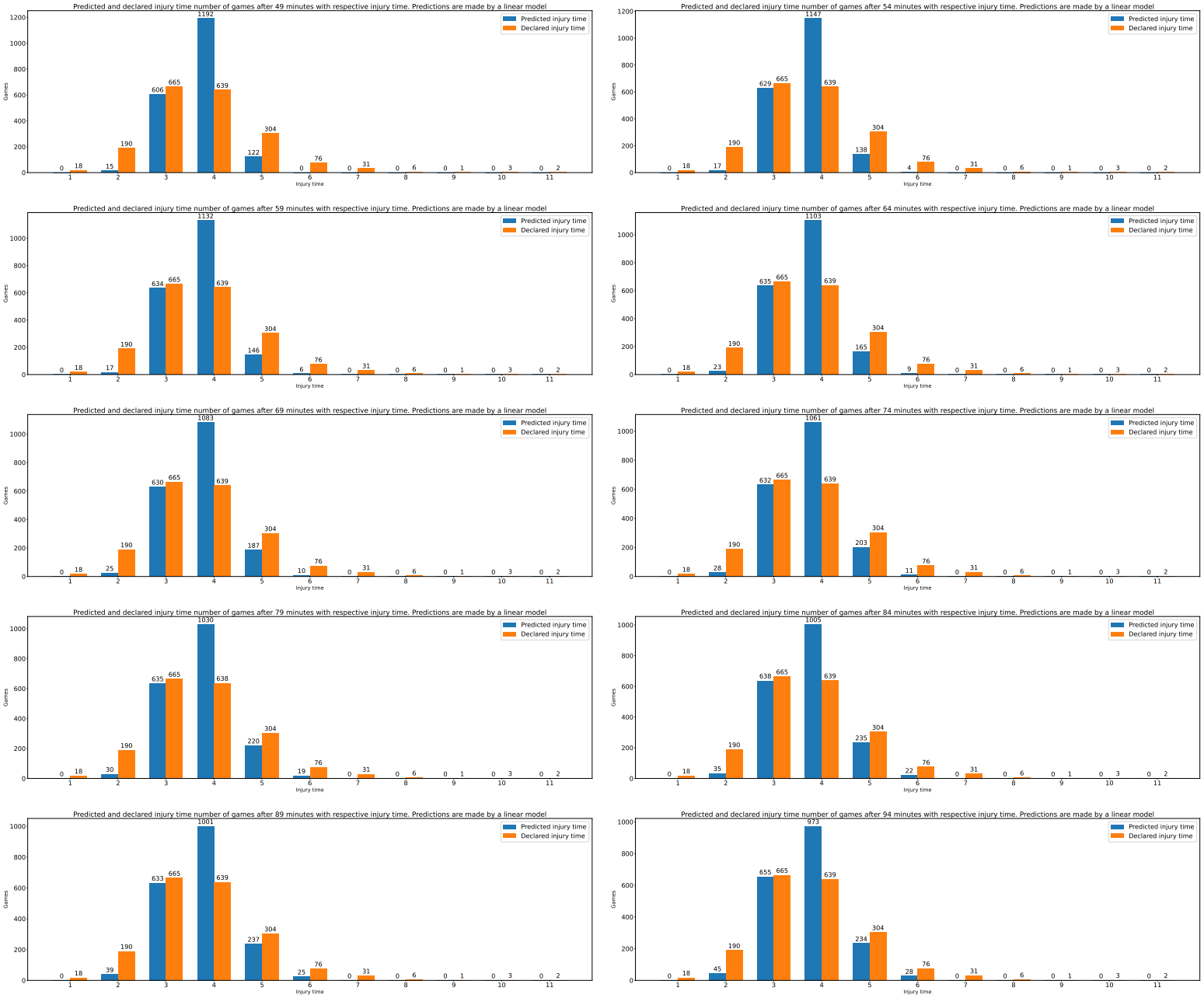


Figure 5.23: Distributions of predicted injury time and declared injury time at every time step

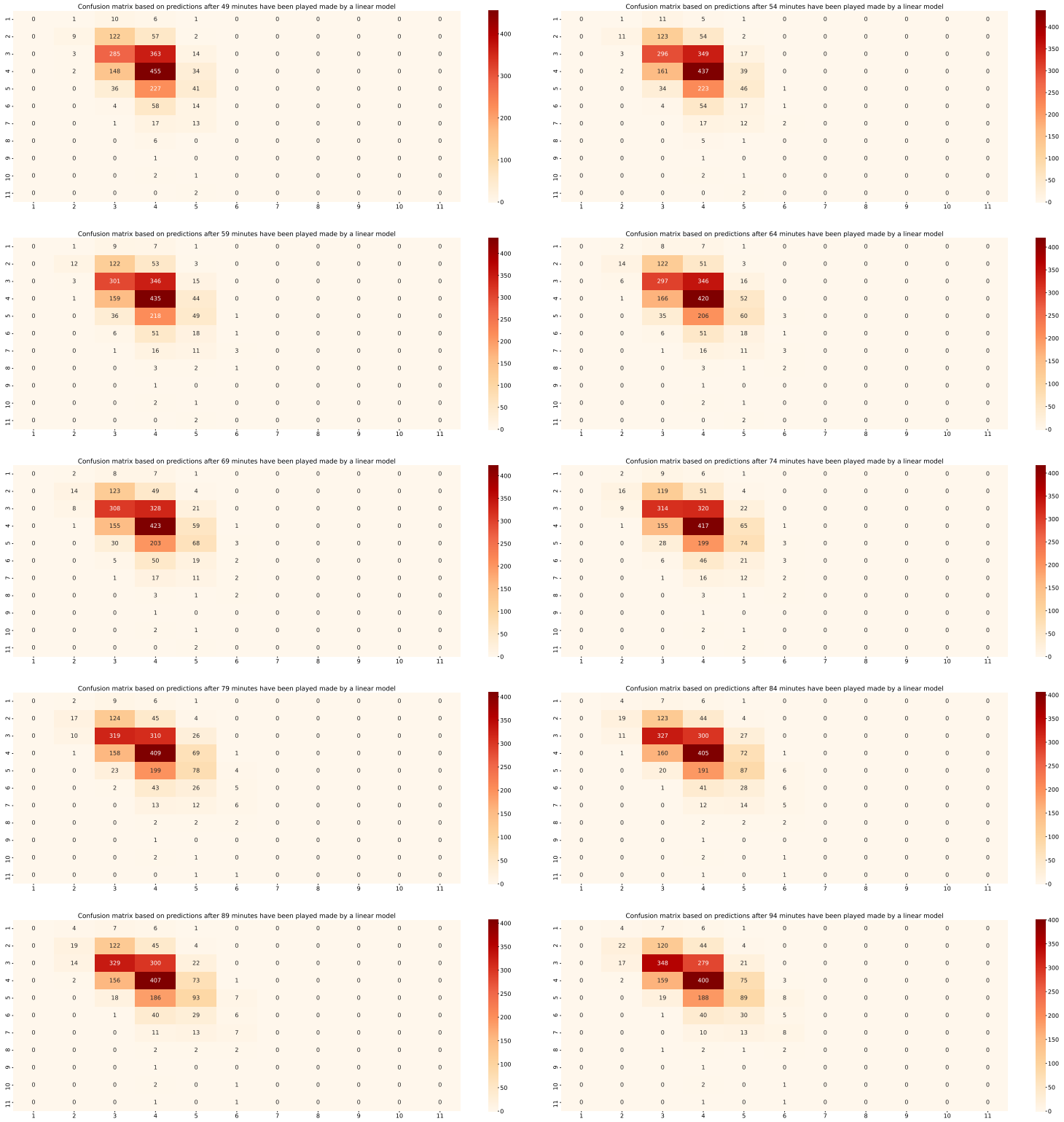


Figure 5.24: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

5.2.2 Poisson model

The null hypothesis for these models are the same as with the pregame Poisson models: There is no significant difference between the actual declared injury times and the predicted injury times by a fitted Poisson model.

First half

Figure 5.25 shows the performance of the Poisson model for real time prediction in the first half. The performance is measured using the same units as before, MSE, RMSE, MAE, MAPE, accuracy and a χ^2 goodness of fit test. Comparing the values for error, the errors are smaller in the real time prediction at every time step and the error decrease with time. The accuracy is starts lower than the pregame prediction, but with time it increases to higher than 0.50. Figure 5.26 shows the expected counts compared to the actual counts, and the means decreases over time, shifting the expected counts slightly towards zero. Figure 5.27 shows the confusion matrix at each time step. At the end of the first half(44:59), the Poisson model most often predicts one minute correctly with an accuracy of 79.69%. The χ^2 result value starts at 5354 and decreases to 245, which means that the fit is much better at the end of the half, than at the start. The p-value however, is never higher than 0.05, and the model is rejected.

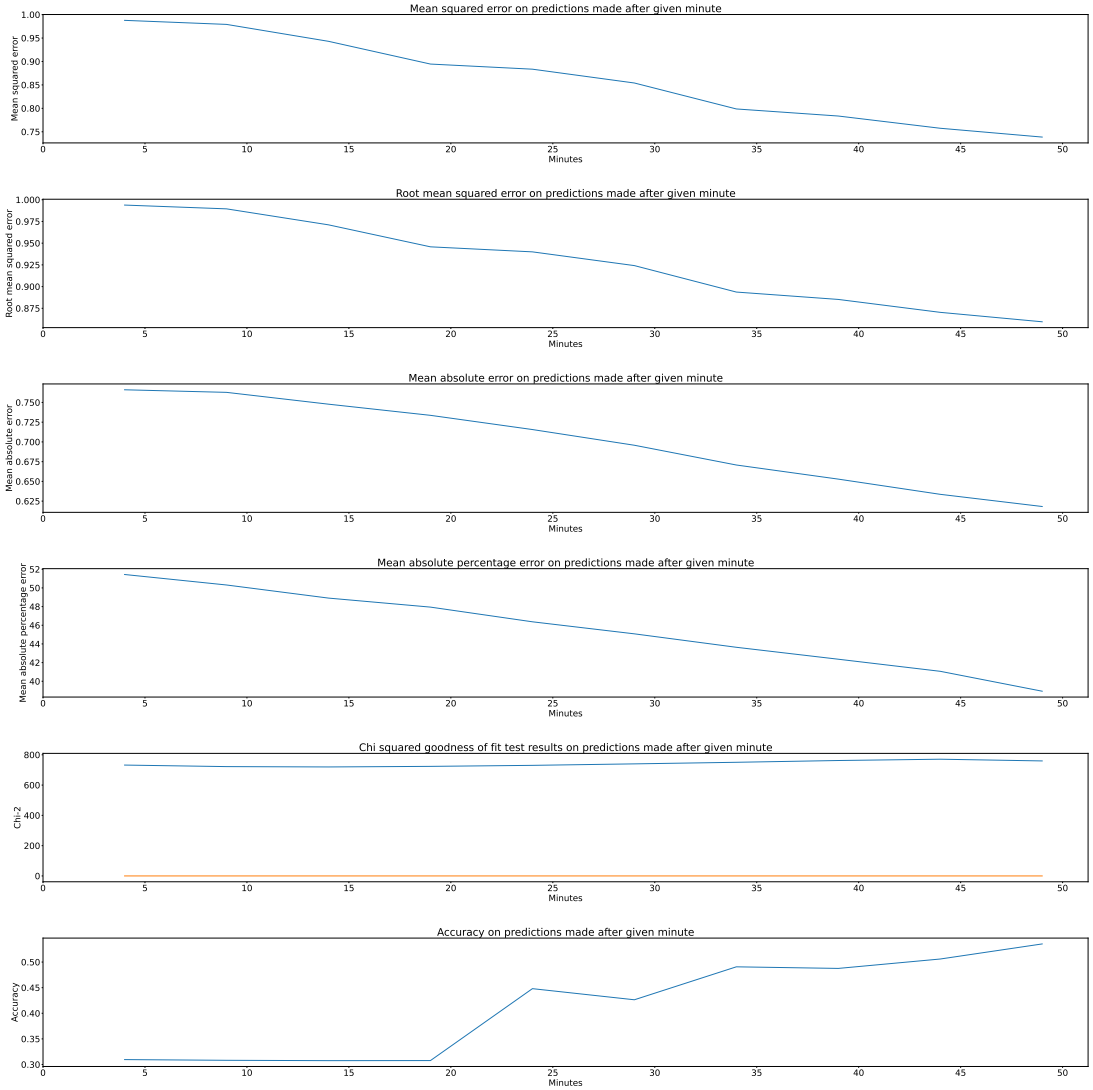


Figure 5.25: Performance on live predictions made by a Poisson model

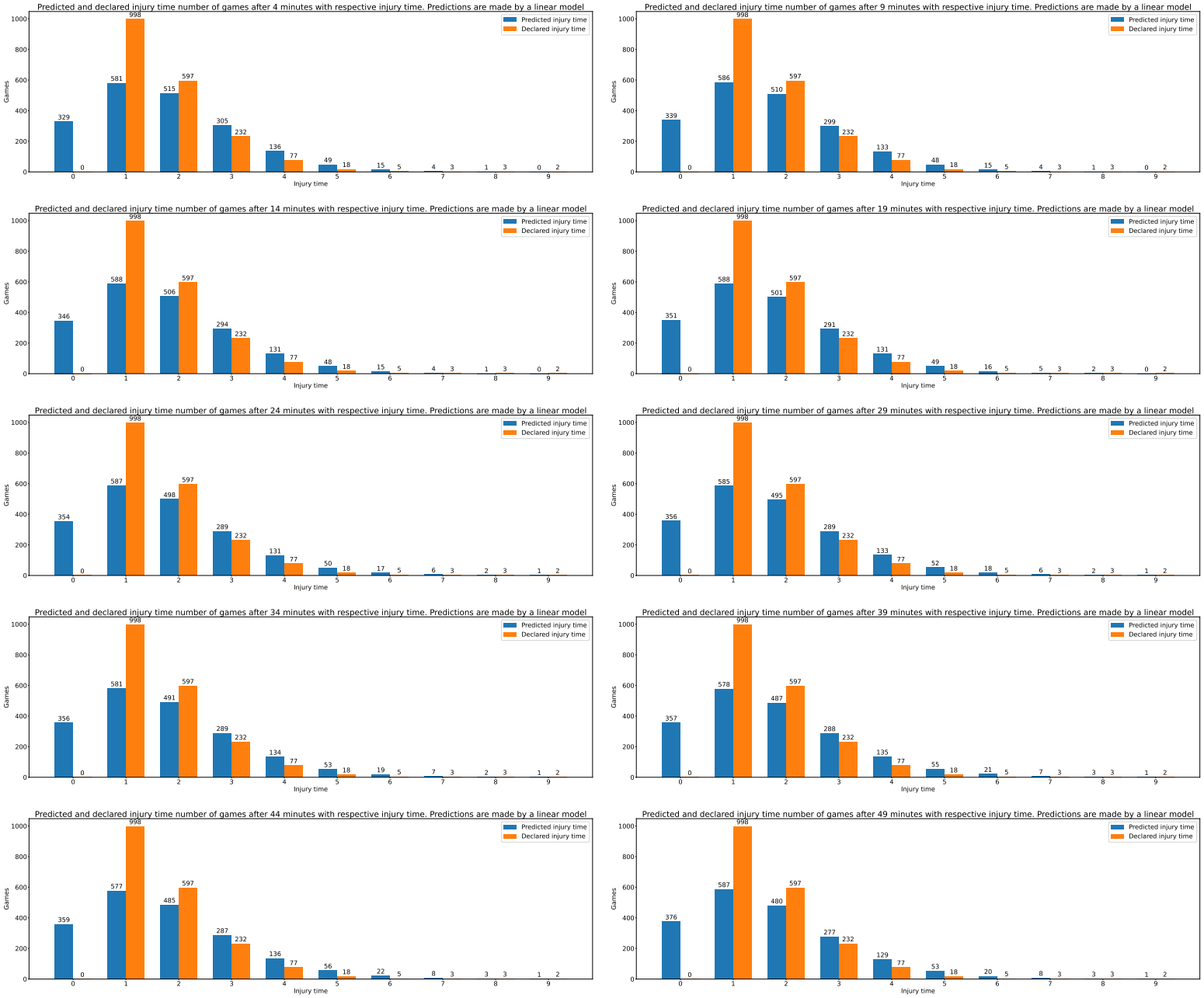


Figure 5.26: Distributions of predicted injury time and declared injury time at every time step

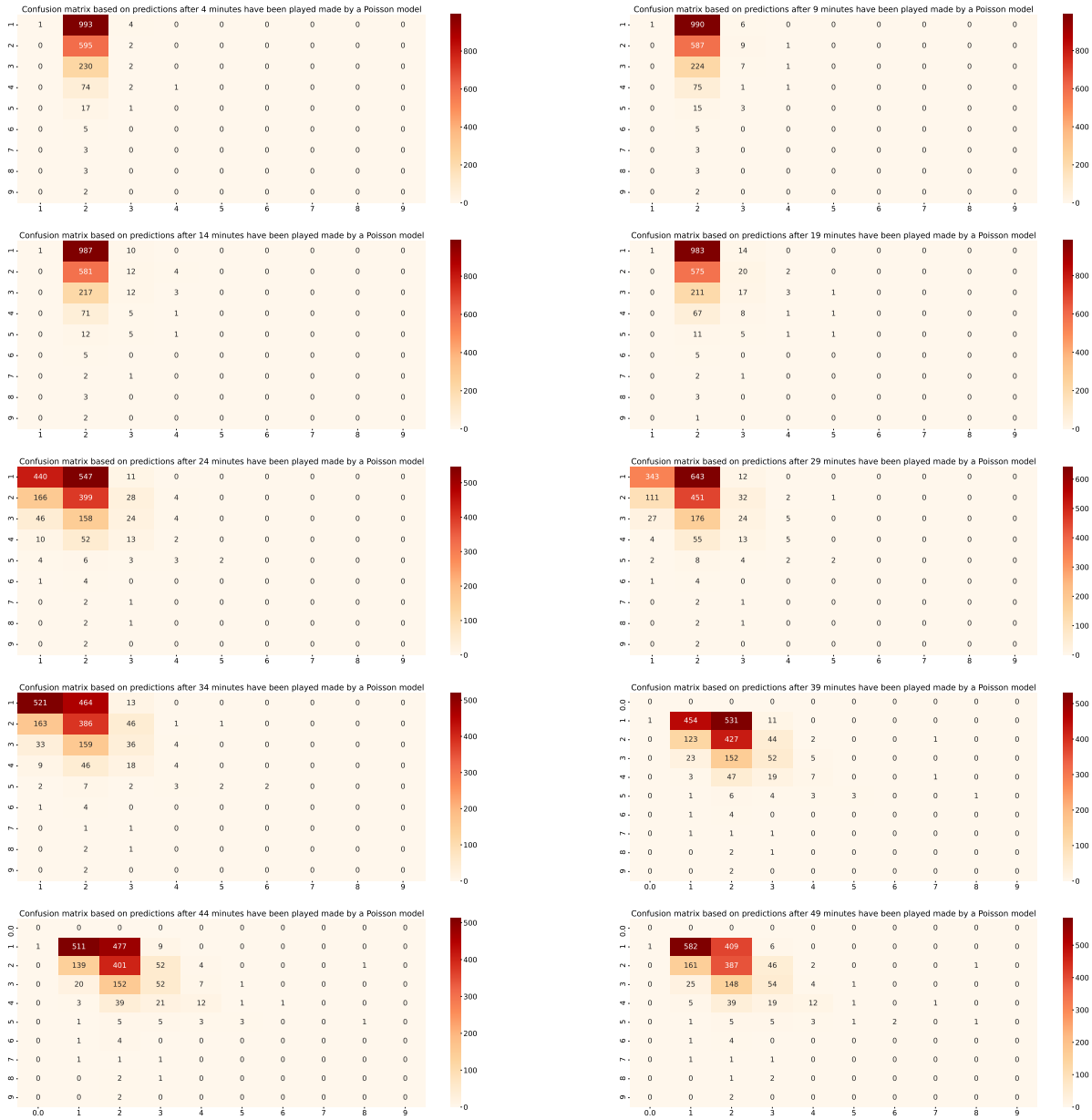


Figure 5.27: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

Second half

Figure 5.28 shows the MSE, RMSE, MAE, MAPE, accuracy and results of the χ^2 goodness of fit test based on the Poisson model used to predict injury time in real time during the second half of a football game. The errors, MSE, RMSE, MAE, MAPE are smaller at all time steps than compared to the before game model. Additionally, the errors decrease with time and the predictions are most accurate right before 4th official puts up the board showing how many minutes are added. In contrast, the accuracy increases, not at all time steps, but in general there is a higher accuracy at the end of the half than at the start. The accuracy is higher after the first five minutes of second half, than before the game and increases up to about 0.445. Figure 5.29 shows the expected frequencies of injury time compared to the actual declared frequencies of injury time. The model predicts a bit smaller means towards as time passes, shifting the distribution slightly towards zero, however this is negligible. Figure 5.30 shows the confusion matrix for every time step. At the end of the second half(89:59), the model predicts most accurately when predicting two minutes, and the accuracy is 66.66%. The results from the χ^2 test, shows that the p-value remains constant below 0.05, meaning there is less than a 5% chance that these number are from the same distribution. Hence the model is rejected.

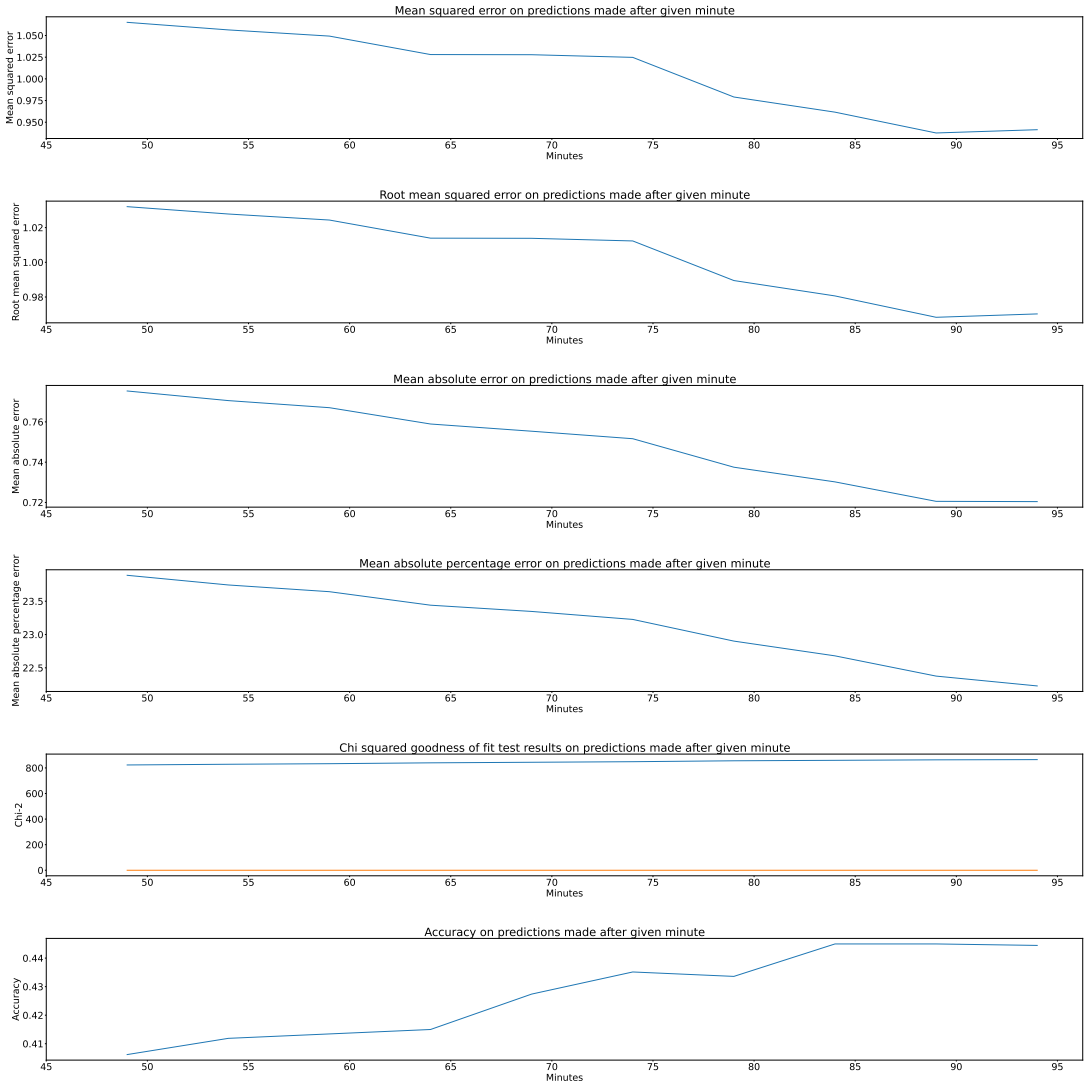
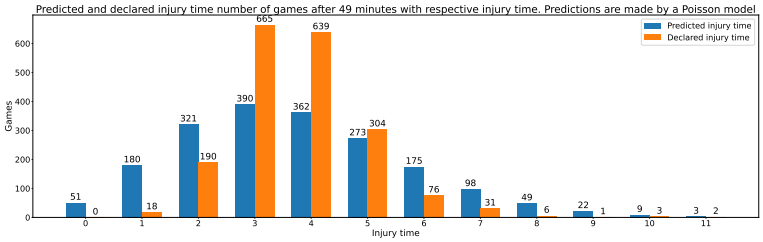
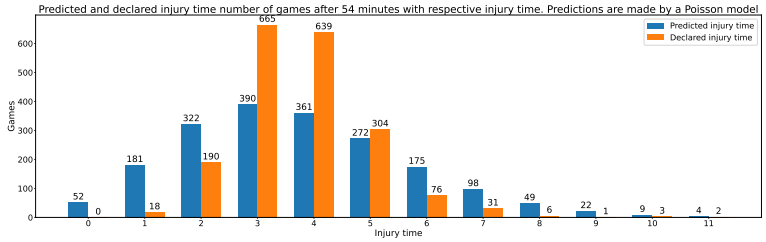
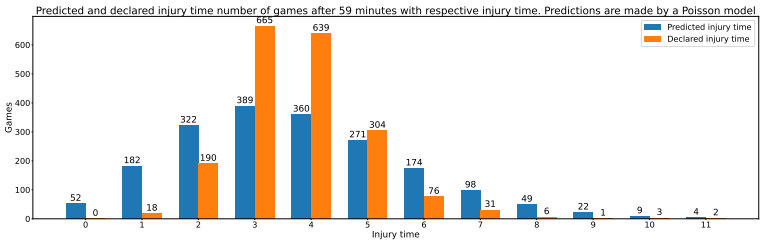
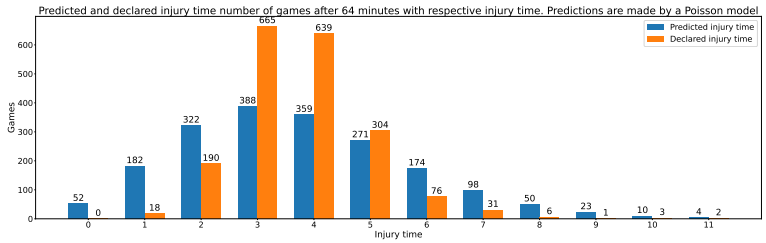


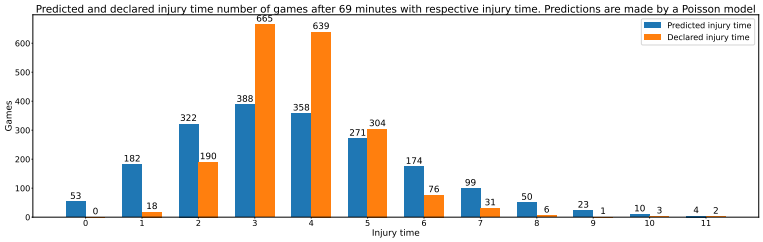
Figure 5.28: Performance on live predictions made by a Poisson model

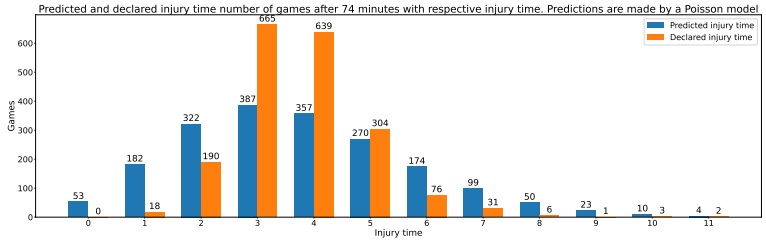


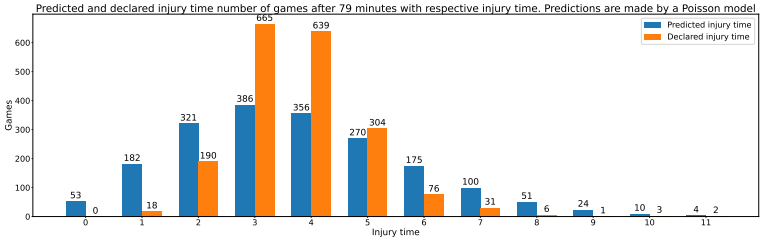


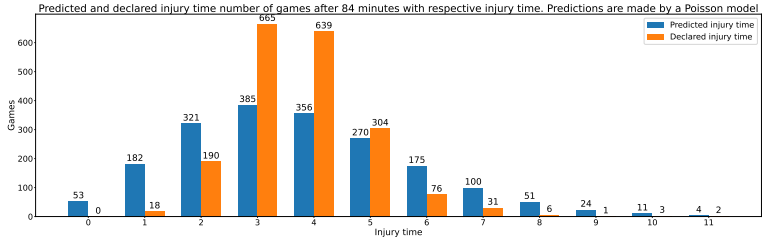


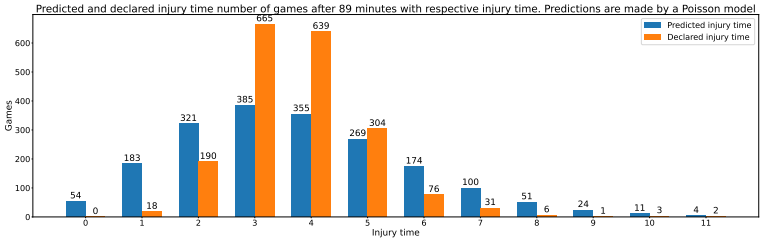












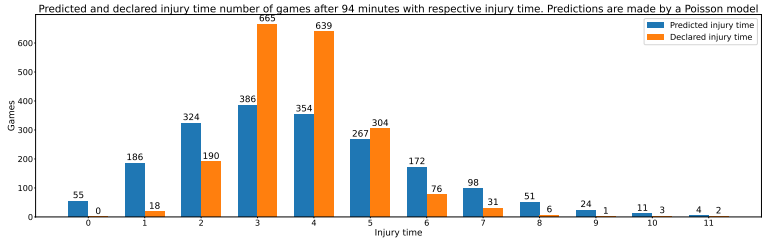


Figure 5.29: Distributions of predicted injury time and declared injury time at every time step

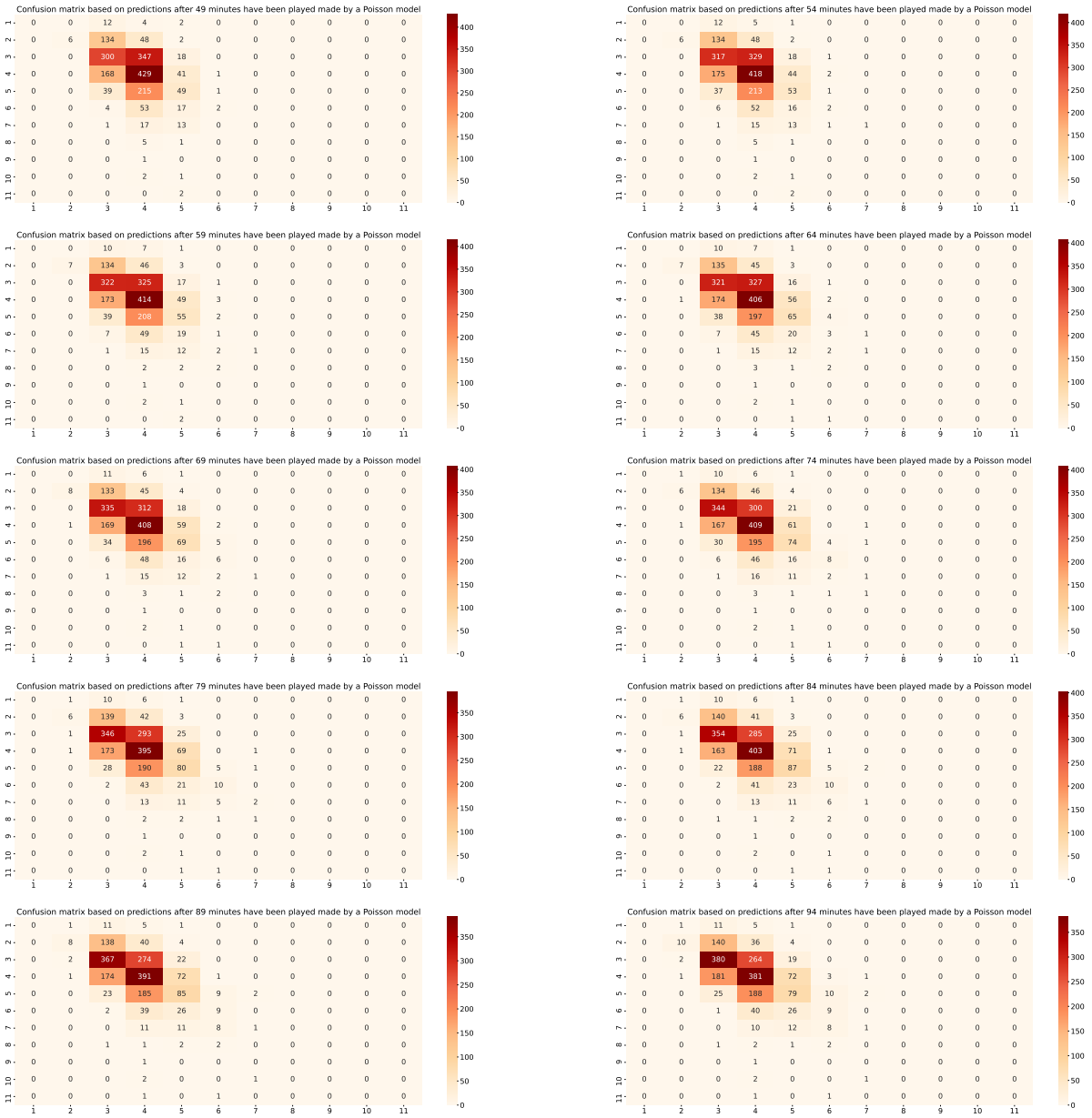


Figure 5.30: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

5.2.3 Negative binomial model

The null hypothesis for the negative binomial models are: There is no significant difference between the actual declared injury times and the predicted injury times by a fitted negative binomial model.

First half

Figure 5.31 shows plots of MSE, RMSE, MAE, MAPE, accuracy and results from χ^2 goodness of fit tests, based on predictions at each time step. The errors decrease as time passes, same applies to the chi^2 statistic. The accuracy, on the other hand increases. Figure 5.32 shows the expected frequencies of the model compared to the actual frequencies. The model, similar to the pregame NB model successfully truncates zero predictions. Figure 5.33 shows the confusion matrix at every timestep. At the end of the first half (44:59), the model has predicts most accurate when predicting one minute, with an accuracy of 79.69%. The results from the χ^2 goodness of fit test shows that the fit improves with time, however the p-value is below 0.05. Hence the model is rejected at all timesteps.

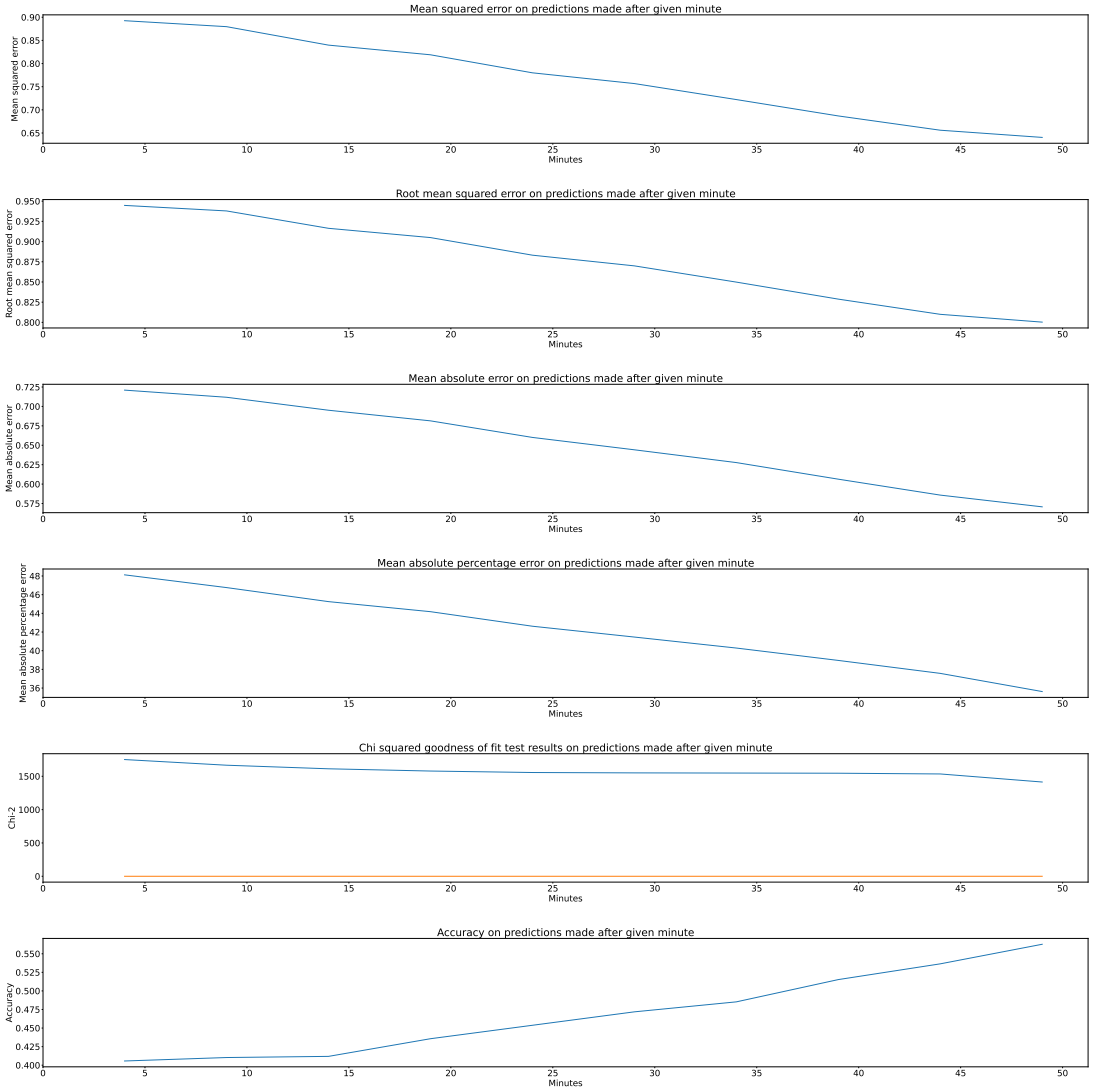


Figure 5.31: Performance on live predictions made by a negative binomial model

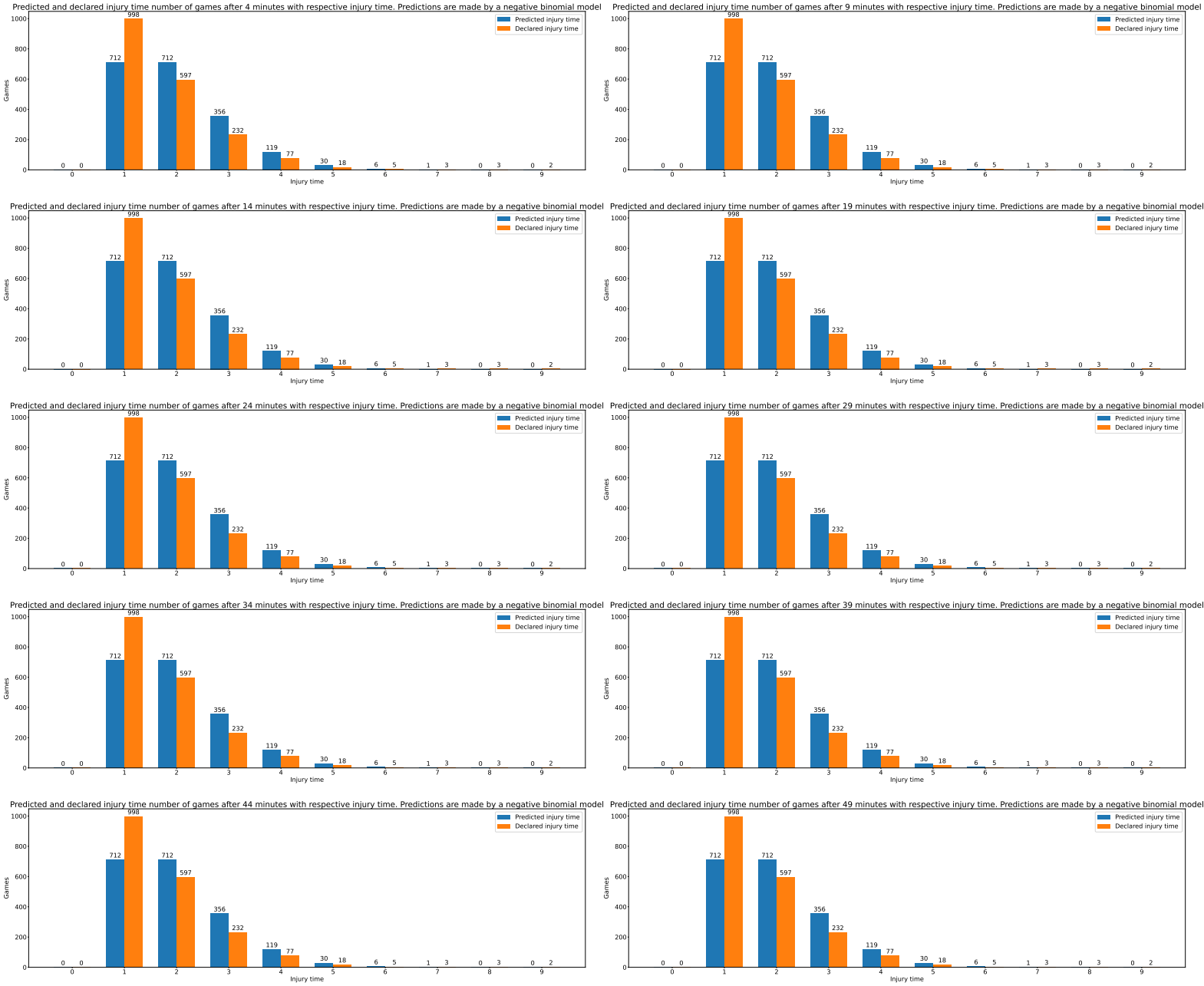


Figure 5.32: Distributions of predicted injury time and declared injury time at every time step

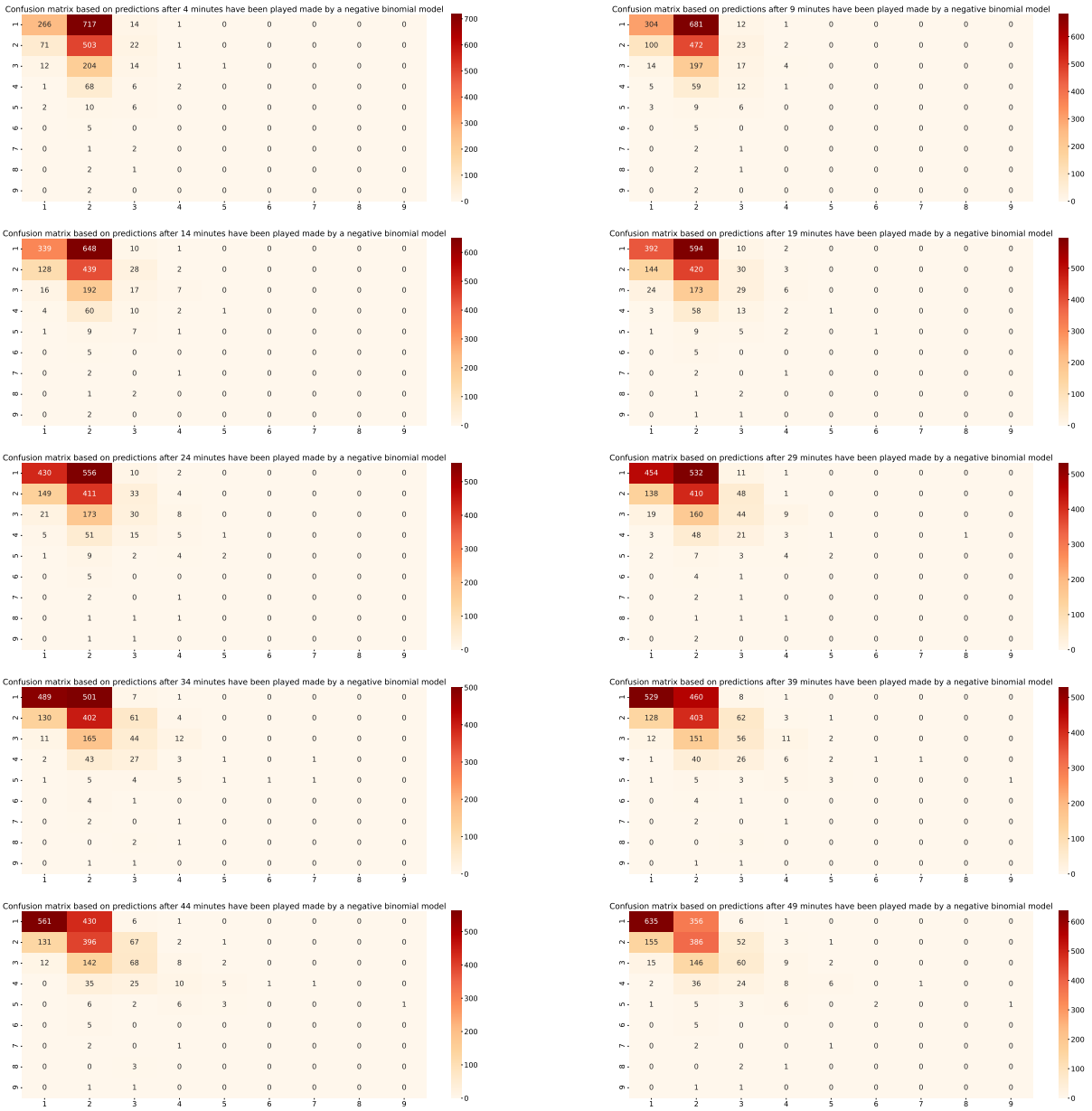


Figure 5.33: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

Second half

Figure 5.34 shows plots of MSE, RMSE, MAE, MAPE, accuracy and results from χ^2 goodness of fit tests, based on predictions at each time step. The errors decreases with time, while the χ^2 statistic remains somewhat constant. In contrast, the accuracy increases with time. Figure 5.35 shows expected frequencies compared to actual frequencies. The model has much more wide frequencies, and there is a negligible tendency to higher predictions as time passes. Figure 5.36 shows the confusion matrix at each timestep, and the model has the most accurate predictions at the end of the game (89:59) when predicting two minutes with an accuracy of 63.63%. The χ^2 goodness of fit statistic and p-value remains constant with time, and the p-value is below 0.05 at all timesteps. Hence, the model is rejected at all timesteps.

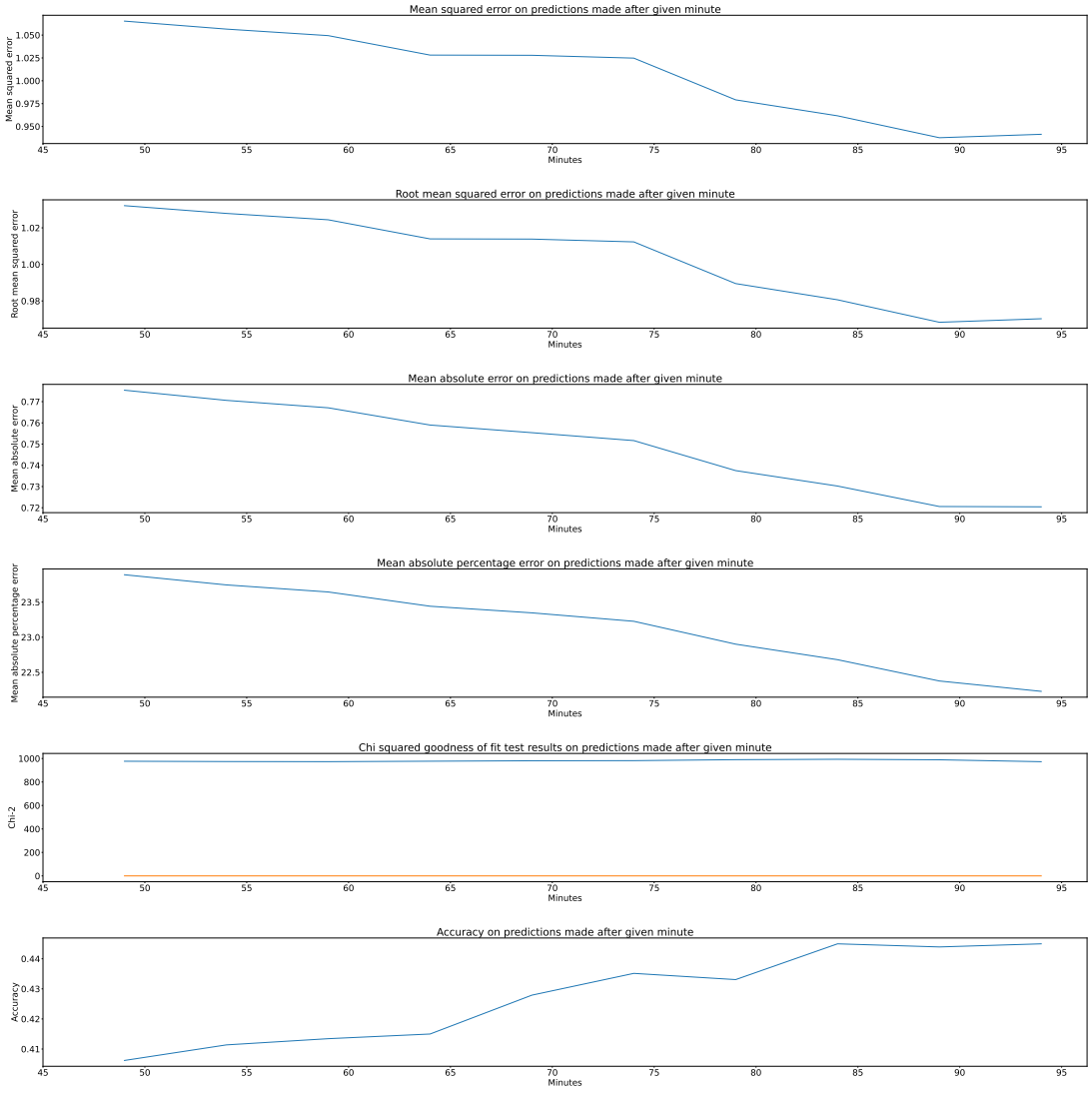


Figure 5.34: Performance on live predictions made by a negative binomial model

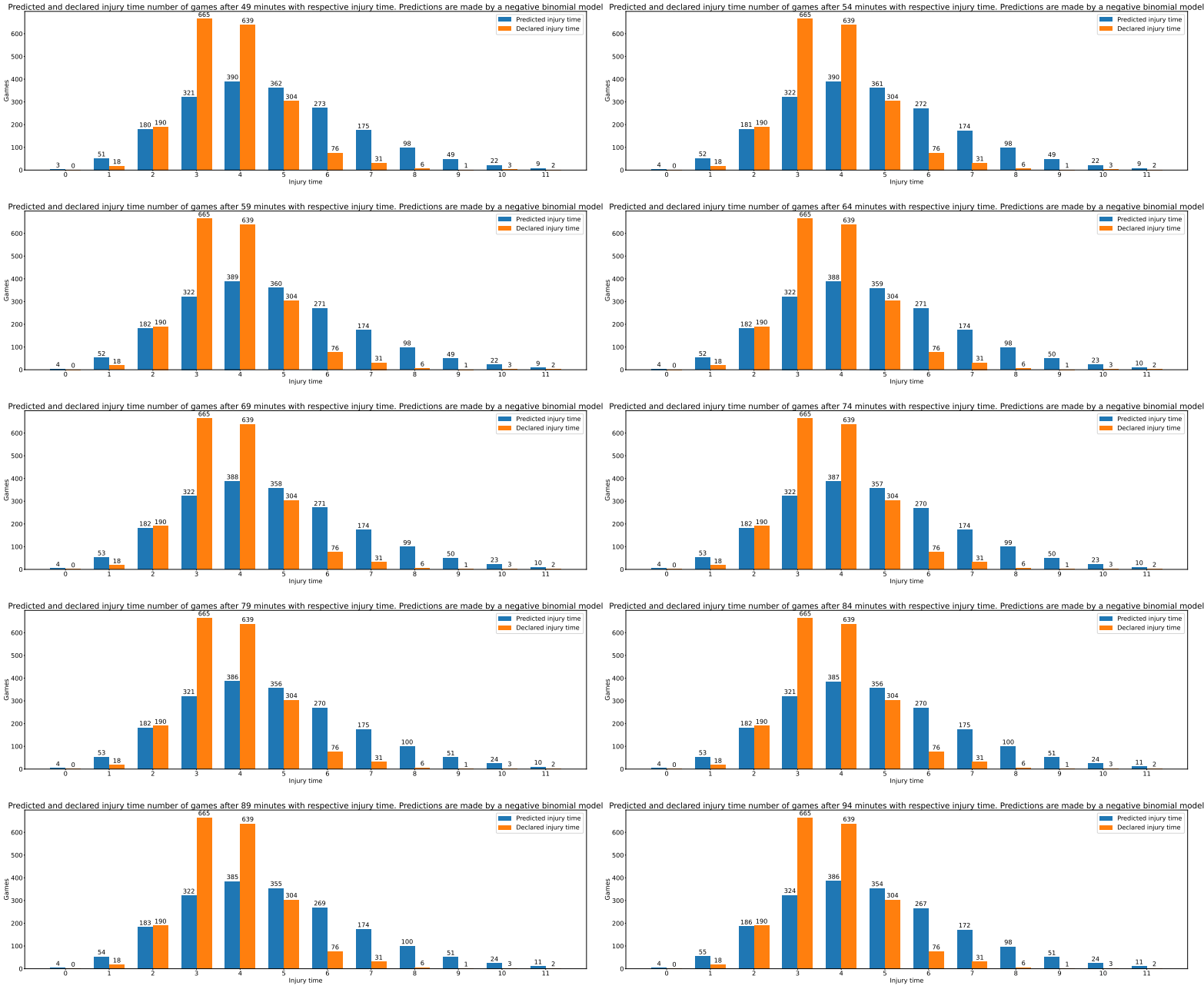


Figure 5.35: Distributions of predicted injury time and declared injury time at every time step

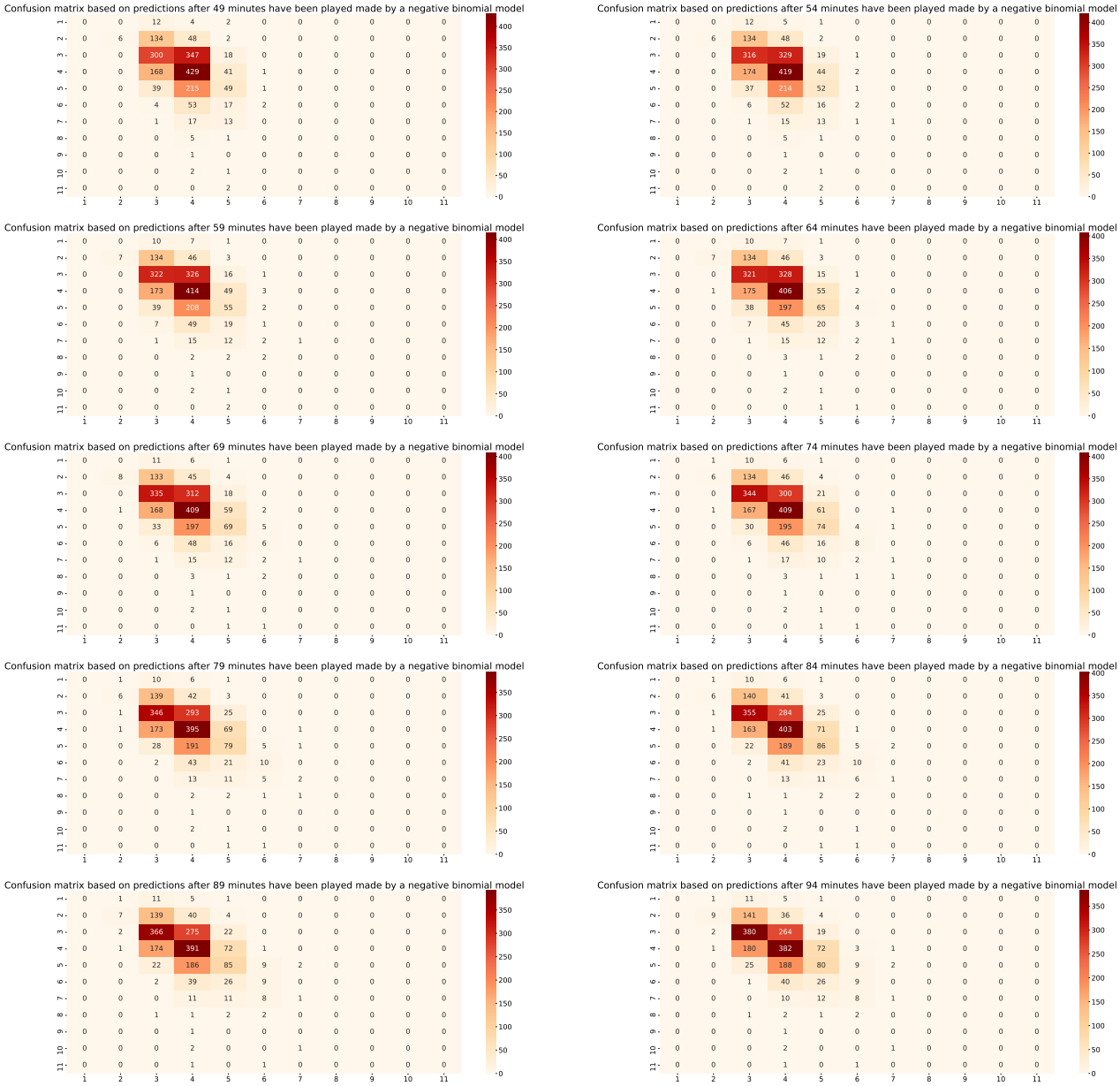


Figure 5.36: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

5.2.4 Regression artificial neural network

A χ^2 goodness of fit test has been completed with the null hypothesis: The declared injury time can be predicted by a regression ANN model without a significant error. The results will either reject or fail to reject this null hypothesis. Both the ANN models has six inputs, goals, substitutions, delay seconds, pregame prediction, start period and end period. Goals, substitutions and delay seconds are cumulative sums of each event. The data is fed every 5th minute, start period and end period is time at the beginning and end of the timestep, the first time step starts at 0 and ends at 4:59. How the rest of the network is built is decided by a hyperparameter tuner made by Keras.

First half

Figure 5.37 is a picture of the resulting neural network after hyperparameter tuning. The network consists of six inputs, six hidden layers with 104, 88, 88, 40, 8, and 8 neurons respectively and one output. For the optimizer a stochastic gradient descent algorithm and the activation function is a Tanh. The model was then trained in 150 epochs and the result for each epoch is saved and the epoch with the lowest error on the validation set, then the network is trained one more time until the best epoch, which was 27.

Figure 5.38 shows the performance of the neural network. The values are smaller than the before model. MAE and MAPE, increases the first few time steps, but after 20 minutes, the errors decrease. Equally for the χ^2 test result, this decreases a lot. Accuracy decreases until the 20th minute and then increases to above 0.50. All of these metrics are connected, when the result of the χ^2 result decreases, this means the fit is better. A better fit, means lower errors and higher accuracy, however the p-value remains below 0,05 and the null hypothesis is rejected. Figure 5.39 shows how many times the model predicts each minute, and at the start of the half the model predicts two minutes almost every time. The predictions varies more with time and the amount of one minute predictions increases from 27 at the 4th minute to 680 at the 44th minute. Figure 5.40 shows the confusion matrix for each time step. At the beginning of the half, the model has the most accurate predictions when predicting one minute with an accuracy of 70.37%. However, the model only predicts one minute about 1.39% of the time. Right before the injury time is acknowledged and the 4th official holds up the board, the model is still most accurate when predicting one minute, with an accuracy of 73.82% and at this time, the model predicts one minute about 35.14% of the time.

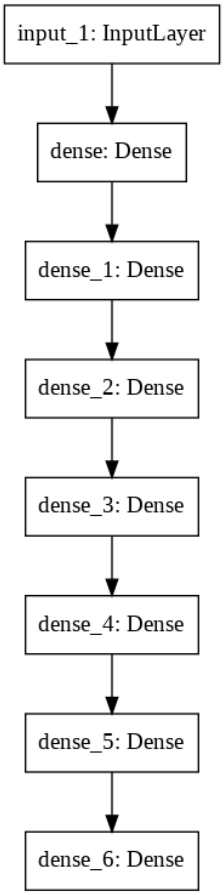


Figure 5.37: The resulting network after hyperparameter tuning with Keras

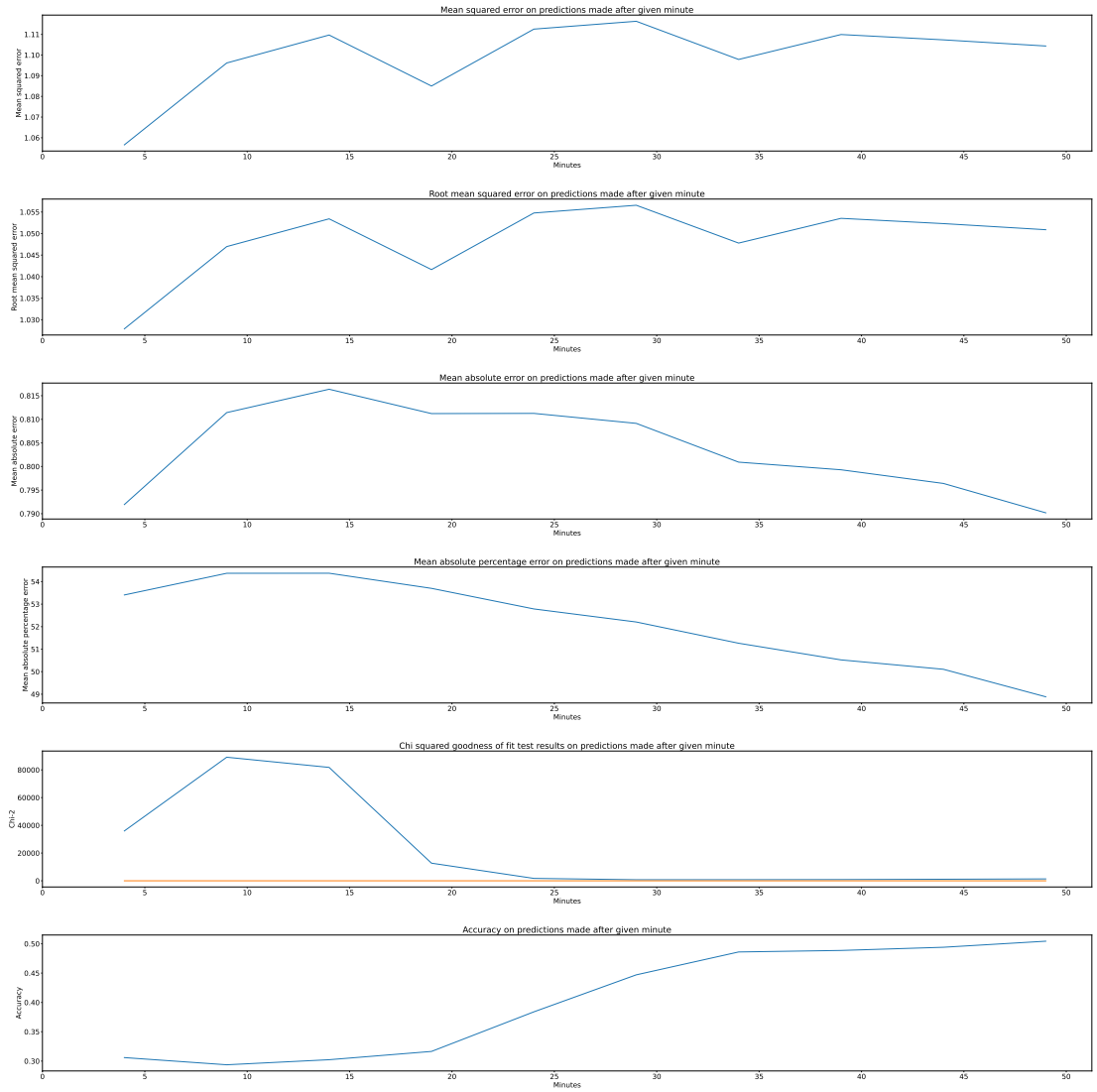


Figure 5.38: Performance on live predictions made by an ANN model

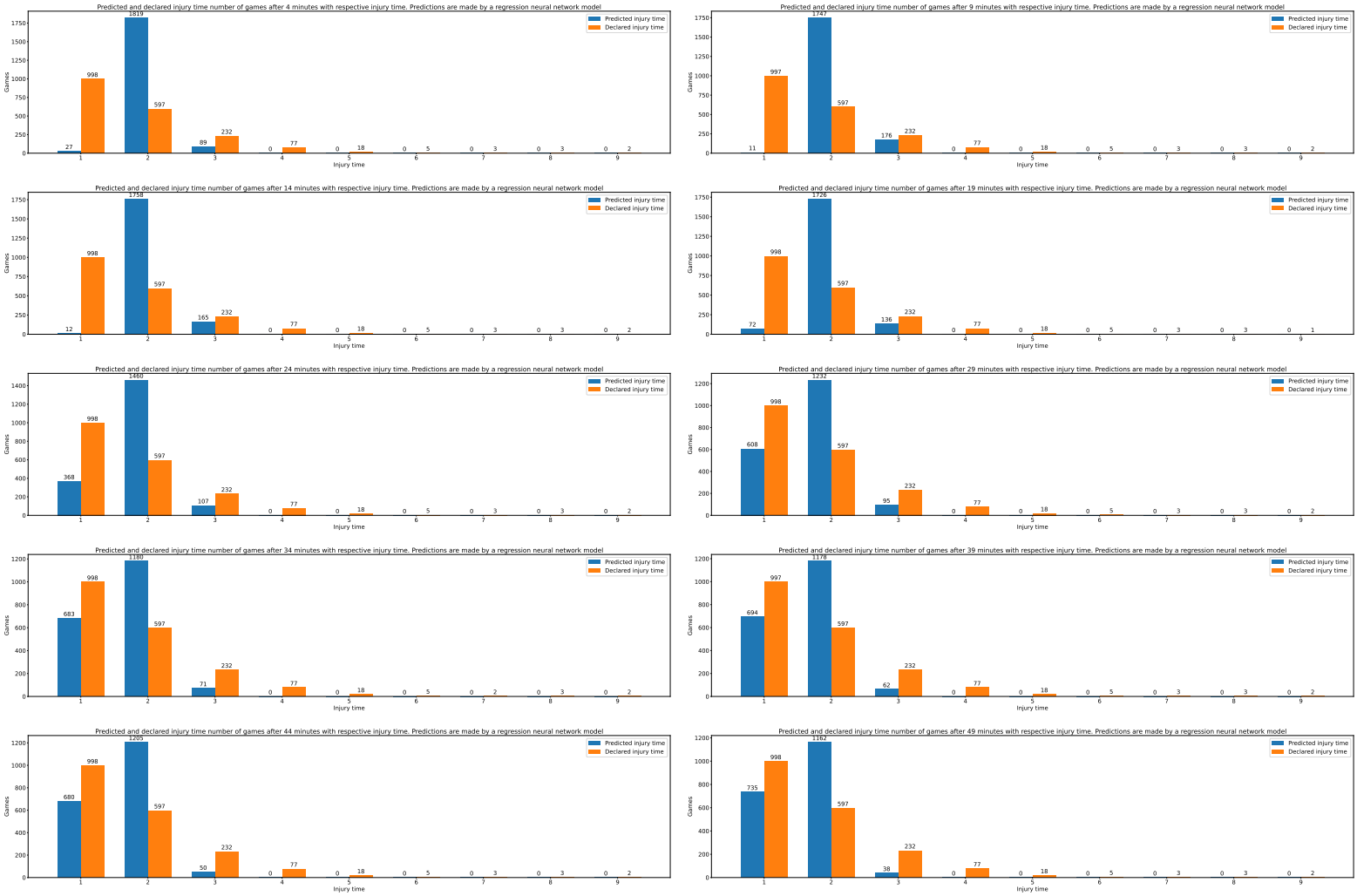


Figure 5.39: Distributions of predicted injury time and declared injury time at every time step

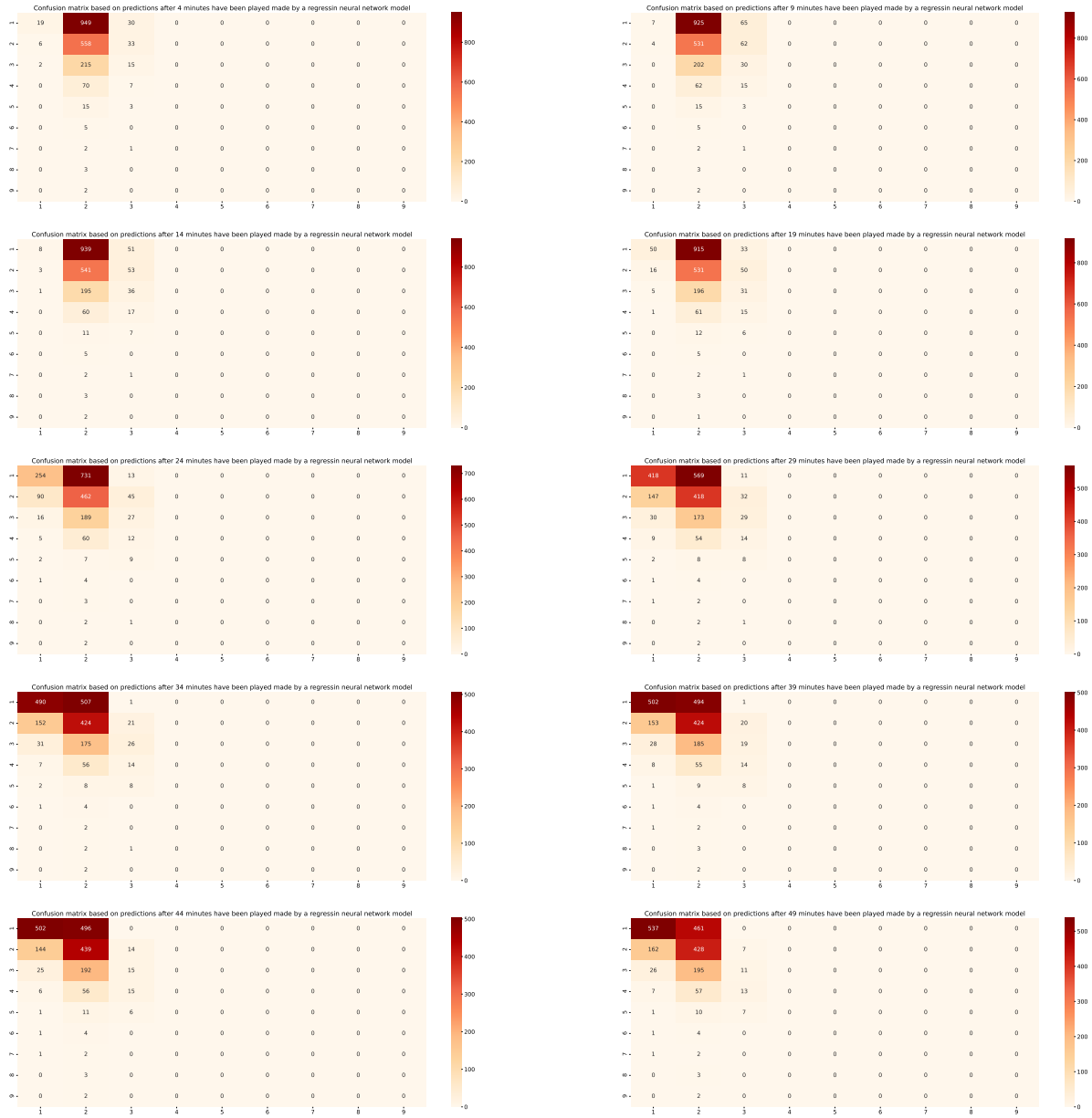


Figure 5.40: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

Second half

Figure 5.41 shows the resulting network after tuning the hyperparameters with Keras tuner. The network consists of six inputs, 4 hidden layers with 8, 120, 104, and 104 neurons respectively and one output layer. For the optimizer Adaptive Moment Estimation (Adam) was selected, this is a computationally efficient algorithm that require little tuning. The biggest difference between Adam and SGD is that the learning rate change during training when using Adam and stays constant when using SGD. The activation function, is equal to the first half network, tanh. Tanh is the most used in recurrent neural networks, however it yielded the best results, hence it was chosen. The training of the network was done in two separate parts, first one round, the network is trained for 150 epochs and the result from each epoch is saved, after this the epoch that provided the lowest error on the validation set was chosen and the network was retrained with 19 epochs.

Figure 5.42 shows the performance of the ANN when used to predict injury time on the test set. MSE, RMSE, MAE, and MAPE at the 49th minute has higher values compared to the before model, and they increase with time. The accuracy is lower after 49 minutes, but increases right above 0.415, which is higher than the pregame model. In contrast the χ^2 test results decrease with time and has a lower value at every time step compared to the before model and with all the other models. On the other hand, the p-value are below 0.05 for all values, hence the null hypothesis is rejected. This model, similarly to the other models, most often predicts four minutes of injury time at the beginning of the half. As time passes, the predictions spreads out. After 49 minutes have passed this ANN model has the most accurate predictions when predicting six minutes, the model only predicts this one time. The second most accurate predictions are when predicting two minutes with an accuracy of 41.91%. After 89 minutes have passed this model predicts most often correctly when predicting two minutes with an accuracy of 46.40%.

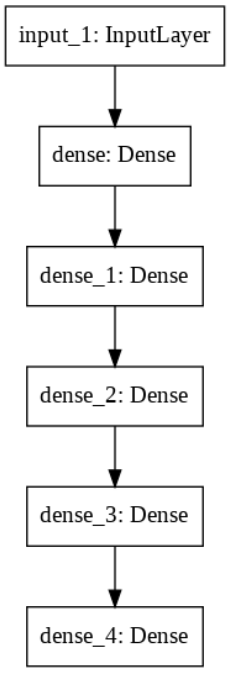


Figure 5.41: The resulting network after hyperparameter tuning with Keras

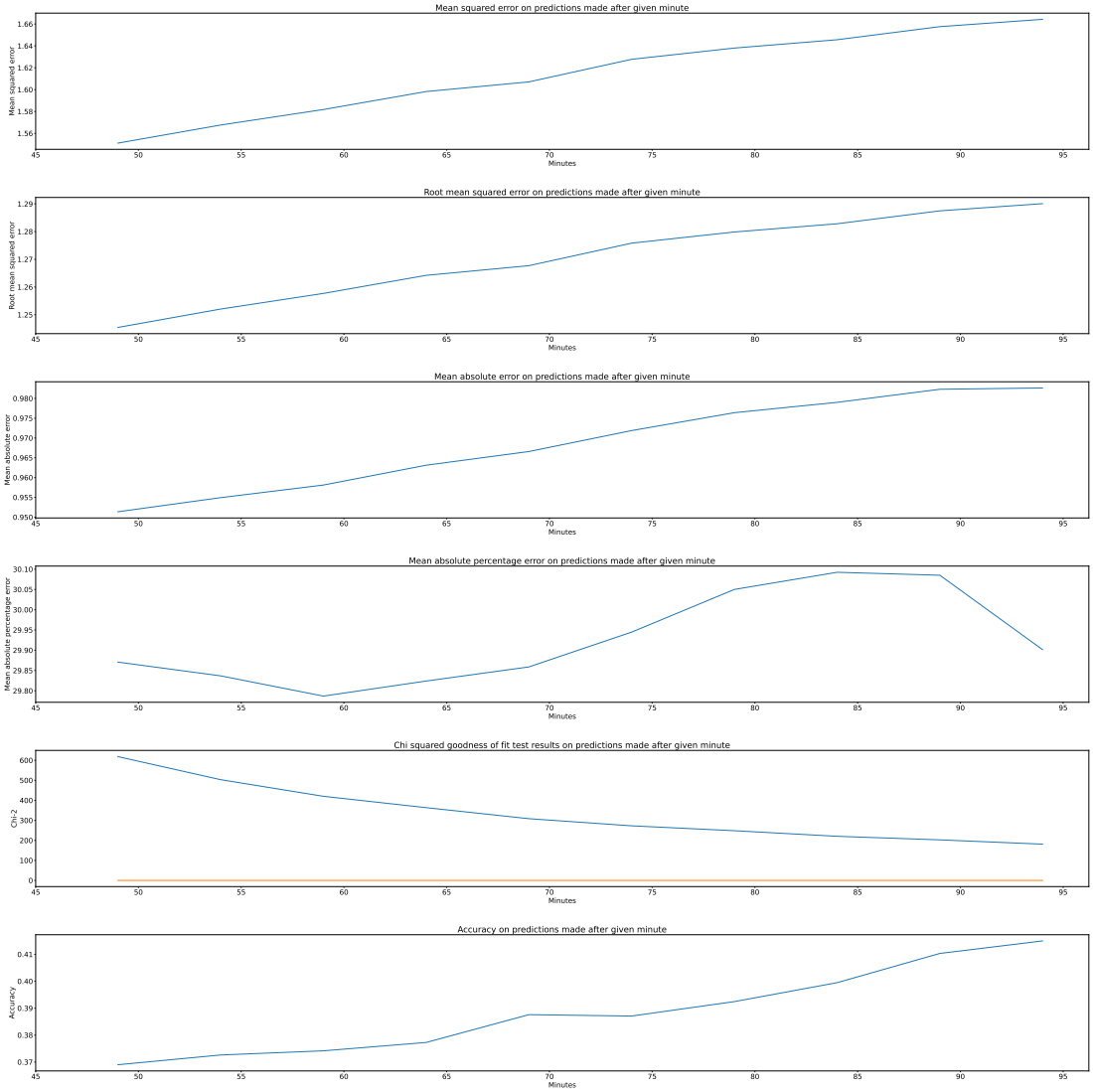


Figure 5.42: Performance on live predictions made by an ANN model

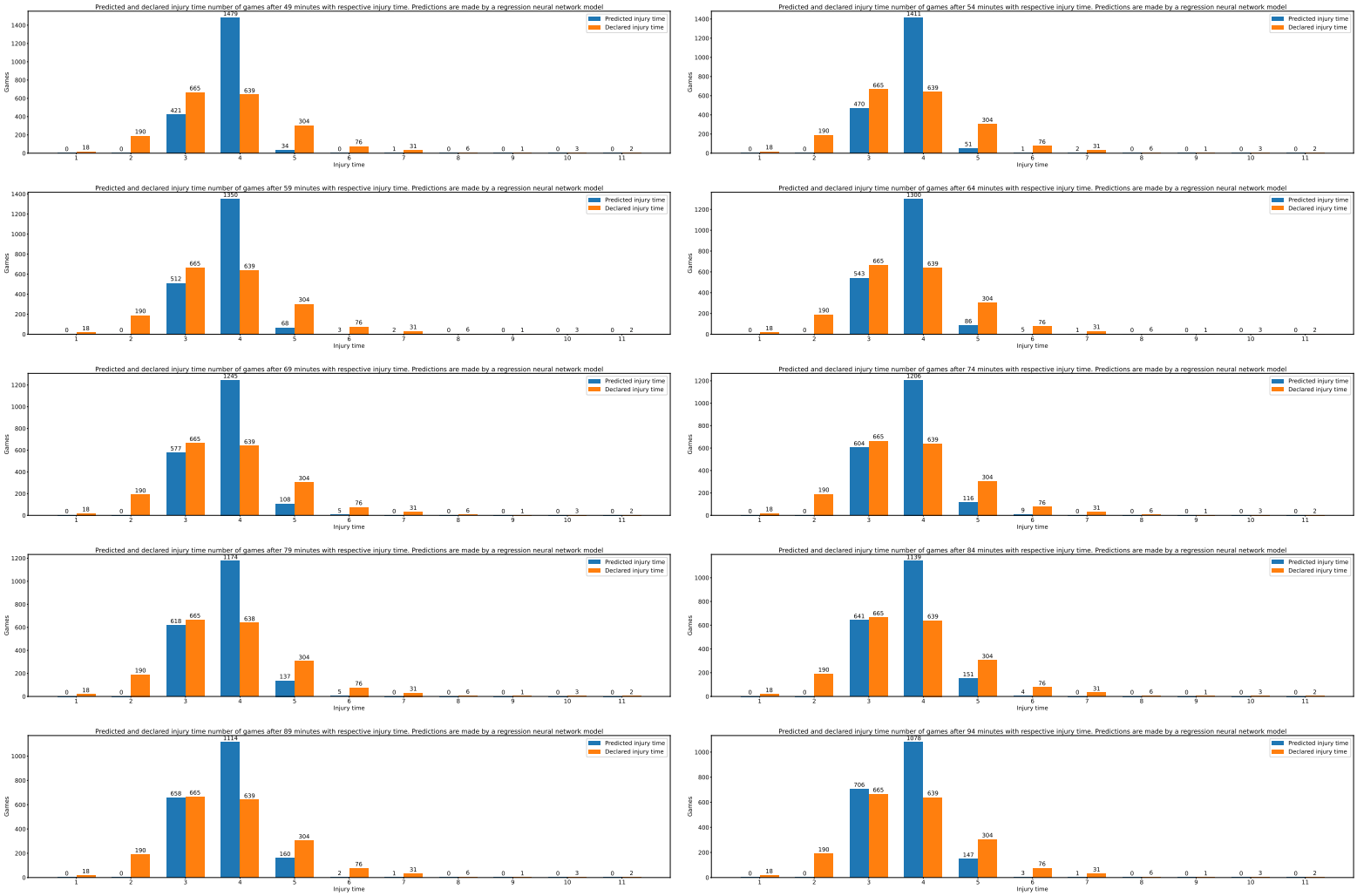


Figure 5.43: Distributions of predicted injury time and declared injury time at every time step

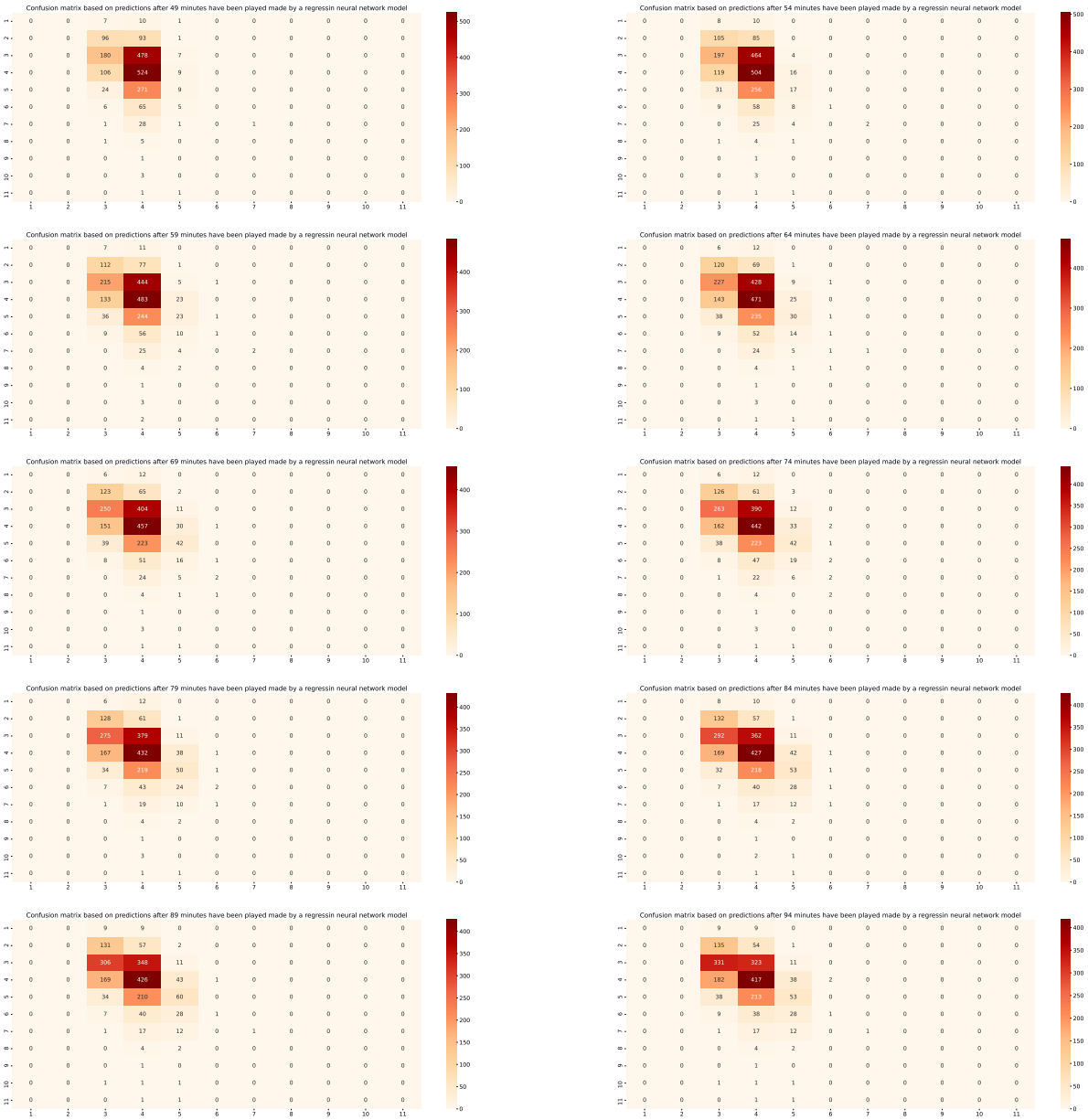


Figure 5.44: Confusion matrices at every time step with predicted injury times on the x-axis and actual injury time on the y-axis

Chapter 6

Discussion and Conclusion

During this thesis several models have been developed and tested in order to predict injury time in a football game. There have been developed separated models for each half and before the game and real time prediction. For every model several error metrics, accuracy and the results of a χ^2 goodness of fit test have been discussed. This section will include further discussion of the results and comparison of all the models.

6.1 Discussion

6.1.1 Discussion of models

For every model a p-value has been calculated based on a χ^2 goodness of fit test, and none of the models have been able to get a p-value higher than 0. The p-value represents a probability of the predicted values based on the distribution of actual declared injury times. Due to limit time, only a few models have been developed during this thesis. The models developed were carefully selected, and in this section the reasoning behind each model will be discussed.

The general opinion regarding injury time is that there is allowed 30 seconds of injury time for each substitution, and goal, and a certain amount for longer stoppages. If so, there would exist a linear equation including the events and the corresponding weights, and a linear regression model would yield good predictions. Based on the results of the linear model, it can be concluded that such guidelines are not followed.

The linear model was rejected. Hence, other models should be developed in order to predict injury time. Injury time is a count variable, and the most common regression model for count variables is the Poisson regression model. Additionally, there have been used Poisson models to predict football scores, with good results, which is why it was natural to explore if a Poisson regression model could yield satisfying results when predicting injury time. However, the results from the Poisson model suggest that injury time is not Poisson distributed.

The Poisson model was rejected, and investigation of the dataset showed that underdispersion occurs. A more robust model, with different mean and variance, is the NB regression model. The NB regression model is also, the second most used count variable regression model, and earlier research shows that this also have been used to predict football scores. However, the results from the NB suggest that injury time does not follow a NB distribution.

The distribution of injury time is unknown, it might not follow any distribution. The linear model, Poisson model and NB model have all been rejected, a model with wider area of use might yield better results. A robust and dynamic machine learning technique is a ANN. When an ANN is constructed appropriately it has a wide area of use and it is very efficient. In the past, neural networks have been used to predict football scores accurately. Additionally, when using a tuner,

there is not much prerequisite required to get satisfying results. However, the ANN model was rejected, and there was a significant difference between the predicted values and actual declared injury time.

The results shows that all the models had to be rejected. When the models are rounded to the closest integer, this introduces a rounding error, which may bias the results. This rounding error exists because of the injury time only can be given in whole numbers. It also applies to the referee, as the injury time can only be in whole minutes. There exists a possibility, that accurate predictions of correct injury time is impossible.

6.1.2 Discussion of data

There are several possible explanations why the models are not able to yield satisfying results. One reason may be the lack of complete data. In all the games where the referee decided not to add injury time, either in one or both halves of the games the data is N/A. This results in 5714 games being removed from the dataset.

The total amount of games, before removal, was 17863, meaning that about 30% of the games were removed. In those 30%, some of the data might be corrupt, for other reasons than zero injury time. However, a sensitivity study to evaluate how the data could look, assuming 25% of the games, 4464 games, had zero minutes of declared injury in the first half, would results in a mean of 1 minute and 16 seconds and 16613 games. As shown in figure 6.2 the dataset includes 6395 games with one minute of added time, and if the mean was reduced from 1 minute and 43 seconds to 1 minute and 16 seconds, most of the predictions would be rounded down to one instead of up to two minutes. Giving each model a higher accuracy.

Including games with zero also would make the data a better fit to a Poisson model, certainly in the first half. Figure 6.1 shows a comparison of the pmf's of a game-independent Poisson model, with mean of 1 minute and 16 seconds, or 1.26, and a game-independent Poisson model with mean of 1 minute and 43 seconds, or 1.72. The figure also compares the density of games before and after adding zeros. Out of the figure, it can be seen that the stipulated "zero-added injury games" probability density function has a much better predictions form than the original with the zero injury time removed. Figure 6.2 shows the expected frequencies of the game-independent Poisson compared to the actual declared injury times. This illustrates the effect of removing larger amounts of data.

Figure 5.5 shows the expected frequencies from the game-dependent trained Poisson model compared to the actual counts. Comparing figures 5.5 and 6.2, the distribution of the injury time predictions is much closer to the observed values, after including zeros, and indicate the importance of having accurate data. As these data are not available, it is not possible to evaluate the impact on the second half and for the real-time predictions. This sensitivity study only includes the Poisson model, but it is expected that with a full, all the models would achieve more accurate predictions.

As depicted in figure 6.1, the probability density functions for the original and the stipulated "zero-added injury time" underneath assuming a Poisson model distribution. Out of the figure, it can be seen that the stipulated "zero-added injury games" probability density function has a much better predictions form than the original with the zero injury time removed.

It seems to be no pattern or tendency between the chosen inputs and the declared injury time. Potential reasons can be explained by statistics in the dataset. These statistics are based on the entire dataset, not just the test set. However, the test set is randomly chosen from the entire dataset and it represents approximately equal statistics. Firstly, in the first half more than half of the games have one minute of declared injury time, however the pregame models tends to predict two minutes. The mean injury time in the first half is 1 minute and 43 seconds, which is an explanation of the two minute predictions. It is worth noting that if one minute was predicted for all games, this would have a higher accuracy compared to the models. The two minute predictions spreads out with time in the real time predictions, however at the end there is still mostly two minute predictions. In the second half three minutes are the most common declared injury time,

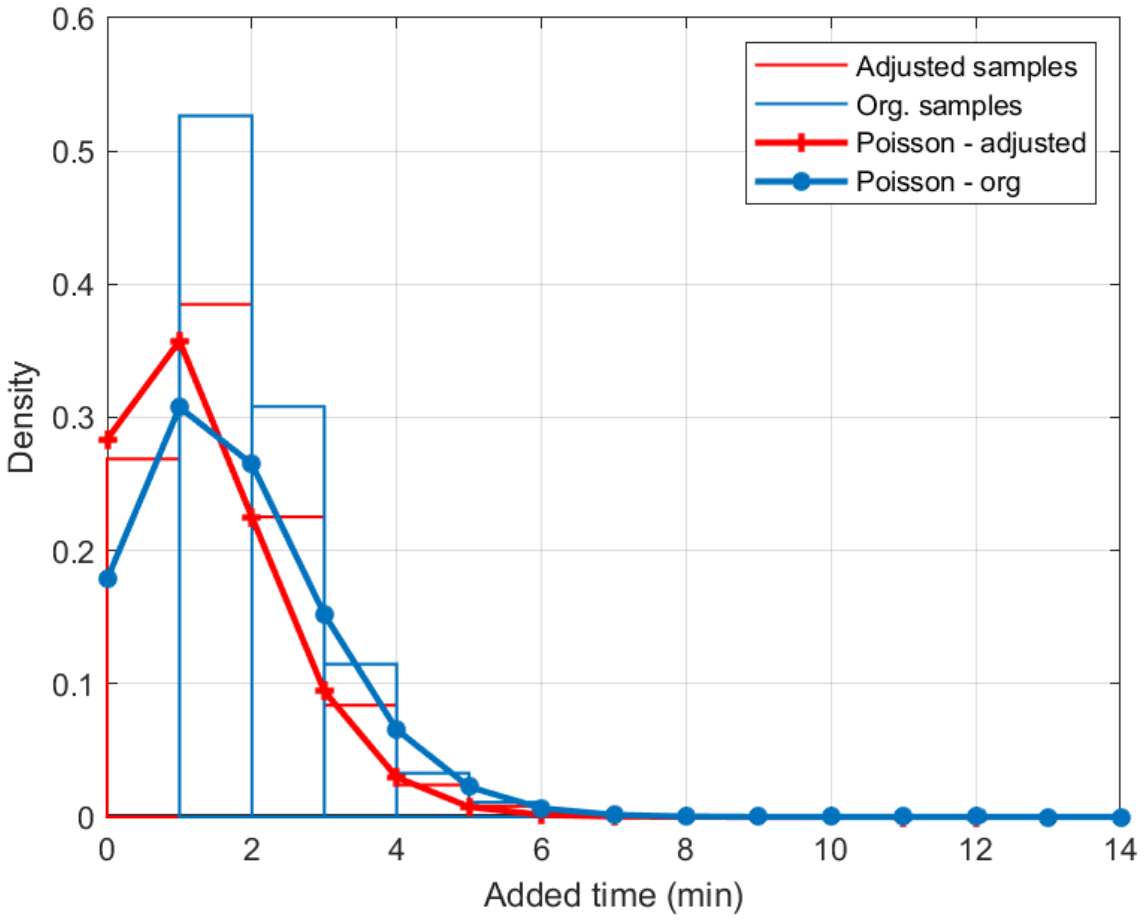


Figure 6.1: A comparison of probabilities from a Poisson model with mean 1.26 and a Poisson model with mean 1.72. Additionally, a comparison of densities of games before and after assuming 25% has zero minutes of declared injury time

the mean however is 3 minutes and 44 seconds. The behaviour of the second half models are equal to the first, the model tends to predict the mean, but the predictions spreads out with time.

The dataset is also missing sanctions, red and yellow cards. It is stated in FIFA law 7 that added time should account for sanctions. How much time that is added due to a yellow or red card is uncertain, due to lack of data. However, as it is included in the FIFA law, it probably would improve the fit of the models. Everything that impacts the injury time should be included in the models, only this way the models will be able to yield accurate predictions. Another uncertainty to injury time is substitutions. These are included in the dataset, and additionally included in the FIFA law, however the dataset does not separate multiple substitutions. Sometimes a team substitutes more than one player at the time, each subsitution is accounted for in the dataset. How this affects injury time is uncertain, a double substitution for example should take less time than two single substitutions, this is not separated in neither the dataset nor models.

The FIFA law states what should be accounted for by injury time, but not how much this must be interpreted by the referee. This introduces bias because every referee will have its own interpretation. Additionally, when there is a known home advantage and referee bias present, this affects injury time. **...** have shown that in close games there is a home advantage when it comes to injury time, if the home team is behind the referees tends to add more additional minutes and if the home team is leading the referees tends to add less additional minutes. One of the reasons for home advantage and referee bias is crowds. **...** documented that the referee bias increases with crowd sizes, until a certain size, after that it stays constant. This also applies to if a bigger team plays a smaller team, if mostly of the crowd supports the away team, then there would be an away advantage. The dataset does not contain any information about crowd sizes, hence including home

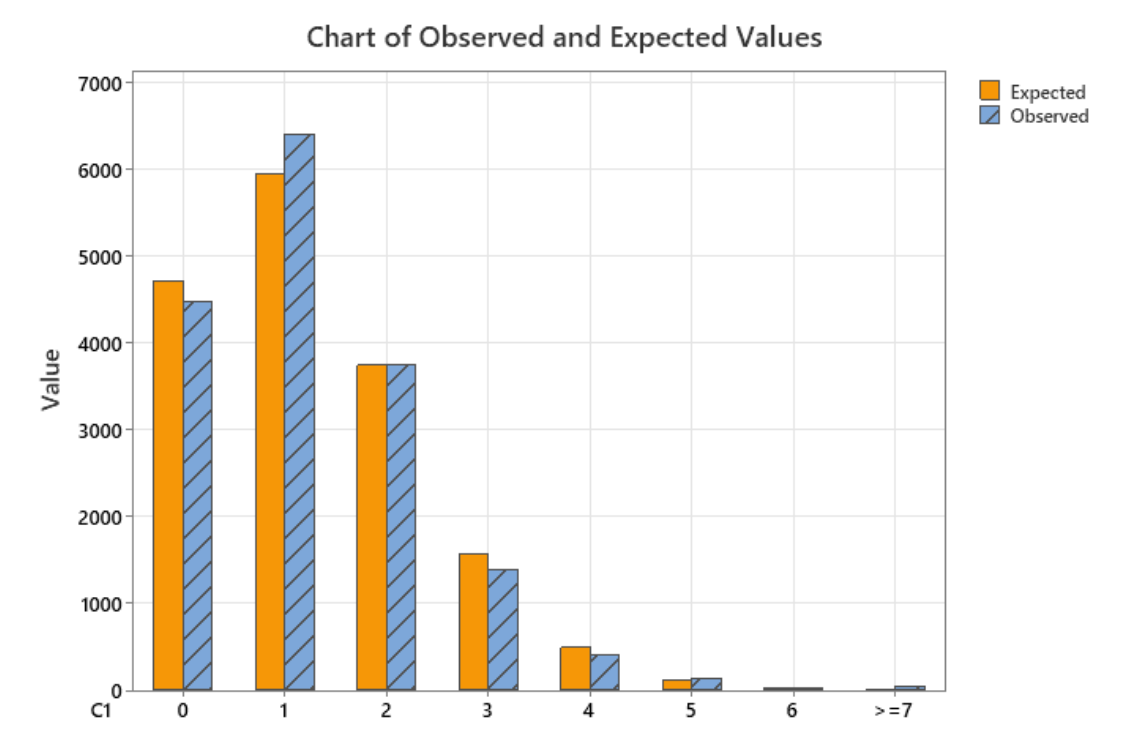


Figure 6.2: A comparison of expected frequencies from a Poisson model with mean 1.26 and the actual declared injury times, after assuming 25% has zero minutes of declared injury time

advantage might do more damage than good and it is not included in the model. Even though it does have a documented effect, it is difficult to make it tangible without crowd sizes, and favorable how many home and away supporters.

6.1.3 Discussion of results

Firstly, the pregame first half models and results will be discussed. The linear model and negative binomial model have almost equal error metrics, when rounded to two decimals. These models have lower error metrics compared to the other models and higher accuracy, but higher score from the χ^2 test. The χ^2 test is a metric of how well the model fit the data and if there is any significant difference between the predicted values and the actual values. In this case, for all the models, there is a significant difference, and all of the models are rejected. The accuracy, a measure of how often the model predicts correctly, is equal for the linear model and the negative binomial model and it is equal for the Poisson model and ANN model. The difference in accuracy is very small, 0.01. On the other hand, the difference in score from the χ^2 is much bigger. The Poisson model and ANN model has much lower result from the χ^2 test. Hence, the differences in errors and accuracy are negligible, and the ANN model is the closest fitted model for the pregame first half predictions out of the models developed during this research.

Secondly, the pregame second half models will be reviewed and compared. Regarding these models, the errors, accuracy and χ^2 test results are opposite of the first half models. For the second half, the ANN model has the lowest errors and highest accuracy, but the highest χ^2 test score. On the other hand, the linear model, Poisson model and negative binomial model all have almost equal errors and accuracy. Similarly as with the first half models, all the models have p-values of zero and are rejected, hence the predicted injury times and actual injury times are not from the same distribution. The model with lowest χ^2 test statistic is the Poisson model, hence this gives the closest fit.

Thirdly, a review of the different first half real time predictions will be conducted. For the real time

prediction in the first half, there is one model which is outperforming the other models at every time step. This model is the negative binomial model, the errors are lower compared to the other models, this model has higher accuracy and the lowest χ^2 test statistic. The negative binomial model has the lowest test score, hence the closest fit. However, the p-values for all models at every time step is much less than 0,05, hence all the models are rejected. Based on the models developed, the negative binomial model yields the closest fit and most accurate predictions of injury time at each time step during the first half.

Fourthly, a discussion about the second half real time models. The errors for the linear model, Poisson model and negative binomial model are very similar, this is similar to the second half pregame predictions. The same applies to the accuracy, all of the models have an accuracy of 0.405 after 49 minutes of play and this increases to just above 0.445. However, the Poisson model increases faster and reaches 0.0445 at the 84th minute, while the linear model does not reach this level before after full time and the negative binomial reaches 0.445 at the 89th minute, right before the 4th official holds up the board indicating how many minutes of declared injury time. Despite equal errors and accuracy, the results from the χ^2 goodness of fit test offers big differences. While the Poisson model and negative binomial model have a χ^2 statistic of about 6000 at the beginning of the half, this decreases to 2348 and 2164 respectively. On the other hand, the linear model has a value around 2500 at the beginning of the half and this decreases to 602, resulting is a closer fit compared to the other models. When comparing the performance of the linear model with the ANN model, the ANN model has higher errors and lower accuracy, but the results from the χ^2 goodness of fit test suggests that the ANN model is a closer fit to the actual declared injury time. The χ^2 test statistic for the ANN model is around 500 after 49 minutes of play and decreases to 110. Hence, even though the ANN model yields higher errors and lower accuracy, this set of predictions are more likely to be from the same distribution as the declared injury times. The p-values for all the models at every time step is below 0.05 and all the models are rejected.

At last, the models will be reviewed in general. Based on the data used in this thesis, all the models were rejected. Improving the data can possibly provide better predictions. However, it is possible that the task of predicting injury time is impossible. Injury time is a integer value, meaning anyway the referee decides to interpret how much time should be added for different events, in different games, there will always be a rounding error. Furthermore referee bias can vary from game to game, meaning there is no pattern in how referee's interpret injury time. In addition, the predictions also suffers from rounding error. Hence, there might not be a trend in how injury time is chosen and it can be impossible to predict.

6.2 Conclusion

During this thesis, both statistical and machine learning models was developed in order to predict injury time in a football game. Predictions have been made both before the game has started, and in real time, every fifth minute, during the game. For each half in a football game, an individual model was developed. The results show that during the game the models has more accurate predictions, but all of the models are rejected by a χ^2 goodness of fit test. Some possible explanations have been discussed in section 6.1, and it is possible that due to rounding error and referee bias, the results achieved is the best possible. However, the models do not provide accurate enough predictions, and they are not trustworthy.

6.3 Future work

Getting correct predictions might be impossible, due to rounding error. A better solution is to use similar models to the Poisson and NB model, which outputs probabilities instead of a point prediction. A model which can be used for this is a multi-class classification ANN, where the input is a game and the output is a set of probabilities for each of the selected classes.

In order to get more accurate predictions of injury time, a good place to start is to get more

accurate data. The dataset lacks all games where, in one of the halves, or both, there was no declared injury time, this has a big impact on the predictions. Another possible addition to the dataset is sanctions, which according to the FIFA law should be accounted for in injury time.

Another suggestion is to gather the odds bookmakers offers on injury time. If it is possible to create an accurate model with better data, this model should be compared against the bookmaker to see if there are any possibilities for value bets.

Further research could be done to see the amount of goals scored in injury time compared to the number of minutes added. According to research done by Pinnacle 2021 in 2014, about 11.3% of all goals is scored during injury time. If it is possible to determine a relationship between how much goals are scored and the amount of time that is added, there are possibilities to find value bets. These value bets would be for example in a game, a model predicts a high number of injury minutes, which again makes it likely to be a high scoring game.

Bibliography

Angelini, Giovanni and Luca De Angelis (2017). 'PARX model for football match predictions'. eng. In: *Journal of forecasting* 36.7, pp. 795–807. ISSN: 0277-6693. (Visited on 3rd May 2021).

Arabzad, S. Mohammad et al. (Oct. 2014). 'Football Match Results Prediction Using Artificial Neural Networks; The Case of Iran Pro League'. In: *International Journal of Applied Research on Industrial Engineering* 1, pp. 159–179. (Visited on 3rd May 2021).

Bengio, Yoshua et al. (2003). 'A Neural Probabilistic Language Model'. eng. In: *Journal of machine learning research* 3.6, pp. 1137–1155. ISSN: 1532-4435. (Visited on 14th June 2021).

Biermann, C. (2019). *Football Hackers: The Science and Art of a Data Revolution*. Blink Publishing. ISBN: 9781788702058. URL: https://books.google.no/books?id=OjMWwQEACAAJ.

Board, IFAB - International Football Association (2021). *Laws of the game*. URL: https://www.theifab.com/laws/latest/the-duration-of-the-match/ (visited on 13th Apr. 2021).

Bordes, Antoine et al. (n.d.). *Learning Structured Embeddings of Knowledge Bases*. eng.

Bunnell, David (2021). *We Timed Every Game. World Cup Stoppage Time Is Wildly Inaccurate*. URL: https://fivethirtyeight.com/features/world-cup-stoppage-time-is-wildly-inaccurate/ (visited on 13th Apr. 2021).

Cheng, Taoya et al. (2003). 'A new model to forecast the results of matches based on hybrid neural networks in the soccer rating system'. eng. In: *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003*. IEEE, pp. 308–313. ISBN: 0769519571. (Visited on 3rd May 2021).

Chollet, Francois et al. (2015). *Keras*. URL: https://github.com/fchollet/keras.

Clarke, Stephen R and John M Norman (1995). 'Home Ground Advantage of Individual Clubs in English Soccer'. eng. In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 44.4, pp. 509–521. ISSN: 0039-0526. (Visited on 13th Apr. 2021).

Colin, Cameron A and Trivedi Pravin (2013a). *Regression analysis of count data, Second edition*. eng. ISBN: 9781107014169.

— (2013b). *Regression analysis of count data, Second edition*. eng, pp. 193–195. ISBN: 9781107014169.

Constantinou, Anthony C (2019). 'Dolores: a model that predicts football match outcomes from all over the world'. eng. In: *Machine learning* 108.1, pp. 49–75. ISSN: 0885-6125. (Visited on 3rd May 2021).

Constantinou, Anthony C, Norman E Fenton and Martin Neil (2012). 'pi-football: A Bayesian network model for forecasting Association Football match outcomes'. eng. In: *Knowledge-based systems* 36, pp. 322–339. ISSN: 0950-7051. (Visited on 3rd May 2021).

Correia-Oliveira, Carlos Rafaell and Victor Amorim Andrade-Souza (2021). 'Home advantage in soccer after the break due to COVID-19 pandemic: does crowd support matter?' eng. In: *International journal of sport and exercise psychology*, pp. 1–12. ISSN: 1612-197X. (Visited on 13th Apr. 2021).

Dixon, Mark J and Stuart G Coles (1997). 'Modelling Association Football Scores and Inefficiencies in the Football Betting Market'. eng. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 46.2, pp. 265–280. ISSN: 0035-9254. (Visited on 3rd May 2021).

FIFA (2021). *2018 FIFA World Cup Russia - Global broadcast and audience summary*. URL: https://www.live-production.tv/sites/default/files/fifa_wc_2018_-_broadcast_audience_summery.pdf (visited on 13th May 2021).

Garicano, Luis, Ignacio Palacios-Huerta and Canice Prendergast (2005). 'Favoritism under Social Pressure'. eng. In: *The review of economics and statistics* 87.2, pp. 208–216. ISSN: 0034-6535. (Visited on 13th Apr. 2021).

Glen, Stephanie (2021). *T-Distribution Table (One Tail and Two-Tails)*. URL: https://www.statisticshowto.com/tables/t-distribution-table/ (visited on 20th June 2021).

Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press, pp. 308–310.

Google (2021). *What is Colaboratory?* URL: https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index#scrollTo=5fCEDCU_qrC0 (visited on 10th May 2021).

Gulli, Antonio and Sujit Pal (2017). *Deep learning with Keras*. Packt Publishing Ltd.

Hardin, James W. and Joseph W. Hilbe (Aug. 2012). *Generalized Linear Models and Extensions, 3rd Edition*. Stata Press books glmext. StataCorp LP. ISBN: ARRAY(0x4f9f0b08). URL: https://ideas.repec.org/b/tsj/spbook/glmext.html.

Harris, Charles R. et al. (Sept. 2020). 'Array programming with NumPy'. In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Hilbe, Joseph M (2011). *Negative binomial regression, second edition*. eng. ISBN: 9780511973420.

Huang, Kou-Yuan and Wen-Lung Chang (2010). 'A neural network method for prediction of 2006 World Cup Football Game'. eng. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8. ISBN: 9781424469161. (Visited on 3rd May 2021).

Hunter, John D (2007). 'Matplotlib: A 2D graphics environment'. In: *Computing in science & engineering* 9.3, pp. 90–95.

Karlis, Dimitris and Ioannis Ntzoufras (2003). 'Analysis of sports data by using bivariate Poisson models'. eng. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 52.3, pp. 381–393. ISSN: 0039-0526. (Visited on 3rd May 2021).

Konaka, Eiji (2021). 'Home advantage of European major football leagues under COVID-19 pandemic'. eng. In: (visited on 13th Apr. 2021).

Lago-Peñas, Carlos and Maite Gómez-López (2016). 'The Influence of Referee Bias on Extra Time in Elite Soccer Matches'. eng. In: *Perceptual and motor skills* 122.2, pp. 666–677. ISSN: 0031-5125. (Visited on 13th Apr. 2021).

Lewis, Michael (2004). *Moneyball : the art of winning an unfair game*. eng. New York.

Maher, M. J. (1982). 'Modelling association football scores'. In: *Statistica Neerlandica* 36.3, pp. 109–118. DOI: https://doi.org/10.1111/j.1467-9574.1982.tb00782.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9574.1982.tb00782.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9574.1982.tb00782.x (visited on 3rd May 2021).

McCarrick, Dane et al. (Aug. 2020). 'Home Advantage during the COVID-19 Pandemic in European football'. In: DOI: 10.31234/osf.io/2gkht. (Visited on 13th Apr. 2021).

McKinney, Wes (2010). 'Data Structures for Statistical Computing in Python'. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

Montgomery, Douglas C., Elizabeth A. Peck and G. Geoffrey Vining (2012). *Introduction to Linear Regression Analysis*. John Wiley Sons, pp. 12–15.

Montgomery, Douglas C., Elizabeth A. Peck and Geoffrey G. Vining (2006). *Introduction to Linear Regression Analysis (4th ed.)* Wiley & Sons, pp. 35–37. ISBN: 0471754951.

Moroney, M.J (1975). *Tall kan tale*. nob. Oslo. (Visited on 20th Apr. 2021).

Nevill, A.M, N.J Balmer and A Mark Williams (2002). 'The influence of crowd noise and experience upon refereeing decisions in football'. eng. In: *Psychology of sport and exercise* 3.4, pp. 261–272. ISSN: 1469-0292. (Visited on 13th Apr. 2021).

Nyquist, R. and Daniel Pettersson (2017). 'Football match prediction using deep learning'. In: (visited on 3rd May 2021).

Pedregosa, F. et al. (2011). 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pinnacle (2021). *How often teams score in stoppage time?* URL: https://www.pinnacle.com/en/betting-articles/Soccer/How-often-teams-score-in-stoppage-time/EKN2MD4ELLUM3RUD (visited on 25th June 2021).

Pollard, Richard (2006). 'Home advantage in soccer: variations in its magnitude and a literature review of the inter-related factors associated with its existence'. eng. In: *Journal of sport behavior* 29.2, p. 169. ISSN: 0162-7341. (Visited on 13th Apr. 2021).

Reep, C, R Pollard and B Benjamin (1971). 'Skill and Chance in Ball Games'. eng. In: *Journal of the Royal Statistical Society. Series A. General* 134.4, pp. 623–629. ISSN: 0035-9238. (Visited on 20th Apr. 2021).

Salvesen, Øyvind (2011). *Statistical models in ice hockey*. eng. Trondheim. (Visited on 3rd May 2021).

Seabold, Skipper and Josef Perktold (2010). 'statsmodels: Econometric and statistical modeling with python'. In: *9th Python in Science Conference*.

Snoek, Jasper, Hugo Larochelle and Ryan P Adams (2012). 'Practical Bayesian Optimization of Machine Learning Algorithms'. eng. In: (visited on 14th June 2021).

Trademates (2021). *Value Bets: How does value occur in Sports Betting?* URL: https://www.tradematesports.com/blog/value-bets-how-value-occurs-in-sports-betting (visited on 13th May 2021).

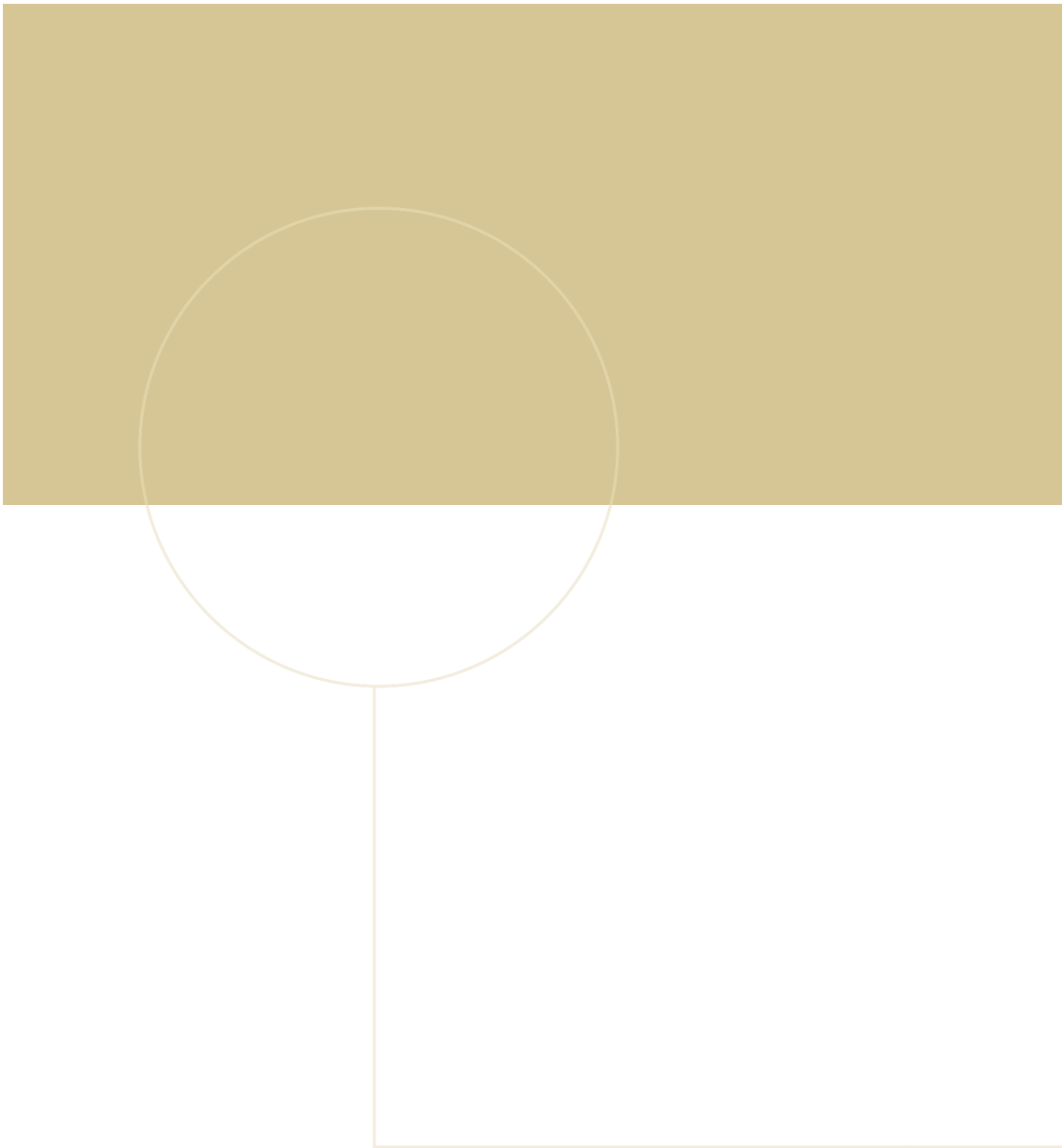
Trainor, Colin (2021). *Additional Time: Spain's Missing Minutes and Other Findings*. URL: https://statsbomb.com/2017/02/additional-time-spains-missing-minutes/ (visited on 13th Apr. 2021).

Vatsvaag, Erik V. (2020). 'Predicting injury time based on events - Project report'. In:

Virtanen, Pauli et al. (2020). 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python'. In: *Nature Methods* 17, pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

Weiberg, Sanford (2014). *Applied Linear Regression*. John Wiley Sons, pp. 62–63.

Zaiontz, Charles (2021). *Confidence and prediction intervals for forecasted values*. URL: https://www.real-statistics.com/regression/confidence-and-prediction-intervals/ (visited on 20th June 2021).



NTNU

Norwegian University of
Science and Technology