Application of a Neural Network to the Shift Scheduling Problem in Supermarkets

Bas Kool
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
b.b.kool@student.utwente.nl

ABSTRACT

The shift scheduling problem in supermarkets has received little attention from current literature, despite it being a recurring and complex task. This research treats this problem as a multi-label classification problem, and proposes a machine learning approach to solve it. An uncomplicated neural network is constructed, which is able to show relatively accurate results compared to the limited amount of implemented features. The research furthermore shows how the current solution can be extended to yield even better results.

Keywords

Neural Network, Multi-Label Classification, Linear Layers, Stochastic Gradient Descend, Shift Scheduling, Supermarkets

1. INTRODUCTION

Personnel scheduling is the weekly recurring task of a planner in which employee schedules are compiled. After the store manager has provided the weekly **steering values**, these are used as the guidelines to construct the rosters, which respect the reported **availabilities** of the employees. Each employee roster consists of one or more **shifts**, together with a set of **tasks** that are expected to be carried out. Each shift inherently states its starting time and duration, and can be extended to include the same information on corresponding breaks. A task can only be assigned to a shift if the associated employee has the appropriate **skill**, and optionally, the appropriate skill level. The personnel scheduling problem is often either defined by minimizing the schedule's computing time whilst meeting predetermined constraints, or minimizing the associated personnel costs whilst realising the expected workload as closely as possible.

The leading steering value in most areas, including retail, is the **expected workload**, determined through demand modelling. These hours are calculated using **WorkForce Management** (WFM) practices. It combines hours calculated using the expected revenue and sold items, with fixed, weekly reoccurring hours, to produce an accurate estimate of the workload. The steering values are often extended to include target **labour costs**, which are linearly dependant on the expected workload and get determined by the manager, since they are not included in the WFM. Demand modelling is followed by **Shift Scheduling**, Days-off Scheduling, Tour Scheduling and Task Assignment. "Shift scheduling involves selecting a set of the best shifts from a (large) pool of candidate shifts on a single day for each employee" [6], forming the daily employee rosters. The exact properties of these shifts and their interrelationships, like the allowance of shift overlap, allowance of overstaffing and understaffing, fixed/variable starting time and length, etc., are dependant on the researched field (f.e. Nursing, Production, Call-center). Days-off scheduling is concerned with determining which days each employee can or can't work, based on law requirements, contract properties, store properties, etc. Tour scheduling combines the previous two processes to construct the week-long rosters. Task assignment then assigns a set of tasks to each shift, completing the schedule. A high level flow chart is provided in Figure 1 to illustrate the interrelationships between these processes.

Even though a lot of progress has been made in different application areas of personnel scheduling, retail has received little attention in current literature [6]. Supermarkets in particular are heavily underrepresented [1, 7], and existing solutions to personnel scheduling problems in other areas cannot be easily converted to apply to supermarkets as well, because of the diversity in constraints and approaches.

Shift scheduling and tour scheduling receive significantly more attention from the literature than the other processes [5], most likely because their results are more directly applicable in the real world. Moreover, shift scheduling is considered the most complex, because of its high number of flexible characteristics [1], like the starting times and durations of the shifts, also making it the area where the most the most progress can be made. This research is focused on the shift scheduling problem as well, incentivized by the need for **optimized rosters** in supermarkets, for which the author had been able to find only a single a single solution [3]. An optimized roster consists of maximized skill at minimal expenses, while honouring employee availabilities. Shift scheduling is thus a both a decision problem, since a shift can only be comprised of **time intervals** matching the corresponding employee's availability, and an optimization problem, attempting to realize the workload while minimizing **personnel costs**.

Literature that does concern a retail environment (or close to), currently proposes constraint programming, often combined with integer programming. The other research on scheduling in supermarkets [3] is an example of this, in which the design of a web-based workforce management

36th Twente Student Conference on IT Febr. 4th, 2022, Enschede, The Netherlands.

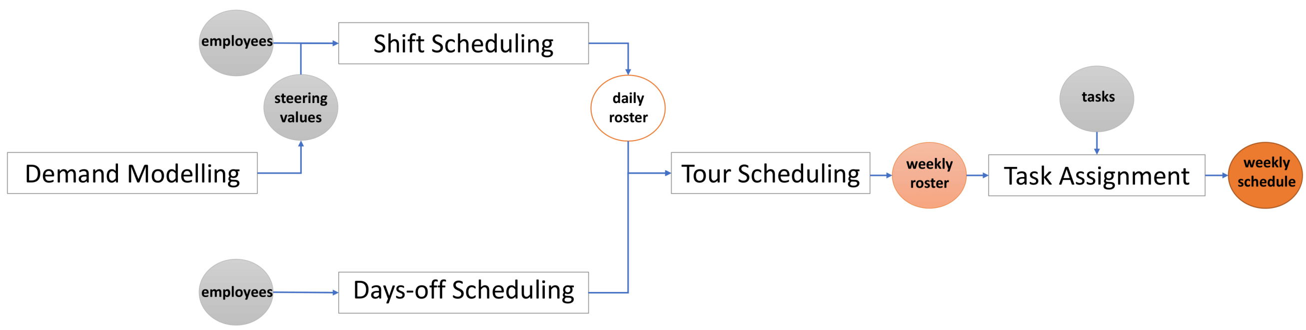

Figure 1. Flow of Personnel Scheduling Processes

system for supermarkets is described, implementing a Branch and Bound optimization algorithm for the personnel scheduling problem. However, this solution type assumes that, to some extent, the optimal instruction set for the creation of optimization algorithms is known. The effectiveness of using conventional machine learning (ML) approaches has not yet been addressed [5], even though state-of-the-art techniques have been proven to generally outperform humans in identifying connections in realised data and constructing an accompanying algorithm. A reason for this could be the complexity and imbalance of the schedules. The decision and optimization problems are transformed into a multi-class multi-label classification problem, in which each employee is assigned to multiple time intervals.

The contribution of this paper is to expand on the existing literature by introducing a Neural Network solution to the shift scheduling problem in supermarkets in the Netherlands. The constructed model serves as a starting point for future research on personnel scheduling in supermarkets.

The paper is structured around the following research question and sub-questions:

RQ How can a neural network be used to solve the shift scheduling problem in supermarkets?

- RSQ1 How is the shift scheduling problem in supermarkets defined?
- RSQ2 How can the shift scheduling problem be defined as a classification problem?
- RSQ3 How accurately can the implemented neural network solve the shift scheduling problem in supermarkets?

In section 2, the shift scheduling problem in supermarkets is further defined, and its different characteristics are stated. Section 3 describes how this problem can be treated as a general classification problem, including evaluation metrics to measure the performance of the solution. Section 4 introduces the model, describing the application a neural network to the research topic, including the datasets and measuring conditions. In section 5, the results of the research are stated and supported by various tables. In section 6, these results are discussed on, and compared to the related work. In section 7, the paper concludes by summarizing the answers to the research questions, and asserting the novel insights that were obtained from the research. It furthermore includes suggestions on how to continue this research.

2. SHIFT SCHEDULING IN SUPERMARKETS

2.1 Flexibility

Shift scheduling in supermarkets has a high degree of **flexibility**. It includes:

- Individual employees
- Full-time employees
- Part-time employees
- Casual employees
- Skills
- Shift starting times
- Shift length
- Breaks
- Shift overlap
- Expected workload
- Understaffing and overstaffing
- Personnel costs

Employees are scheduled individually. Depending on their contract type, they either have recurring rosters, or a weekly changing roster honouring the contractually agreed amount of hours. Skills in supermarkets correspond the the store's departments (F.e. Checkout, Fresh, Bread, etc.), and employee is allocated to one or more departments. Allowing exchanges between departments is optional. The starting times and lengths are definable per shift. Each shift is assigned a lawfully predetermined break length, dependant on the shift starting time and length, and possibly extended by the store manager. Employees are free to choose when to apply the break. Shift overlap is allowed. Determining the expected workload is usually an outsourced process, based on expected revenue, customers and sold items, and expressed in necessary hours per time interval. Understaffing and overstaffing are allowed, though minimized. From the possible rosters that comply to all constraints, the optimal choice is made based on personnel costs.

2.2 Constraints and preferences

As mentioned, the shift scheduling problem is both a decision, and an optimization problem. The decision problem is defined as a set of hard constraints, and the optimization as a set of soft constraints:

Hard constraints.

- Honour employee skills
- Honour contractual agreements
- Honour working hours
- Honour time interval
- Honour recurring rosters
- Honour employee availabilities

- Maximum shifts per day
- Minimum hours per shift
- Maximum hours per shift

Besides honouring the respective store's working hours, all time related constraints are divisible by a predefined time interval, which is usually 15 minutes. The maximum and minimum constraints are lawfully defined, but can be further constrained by the store manager. Break length determination is not part of the problem description, because it is directly dependant on the corresponding shift.

Soft constraints.

- Satisfy staff requirements/expected workload
- Minimize personnel costs

The soft constraints are used to determine optimized schedules from those obtained by applying the hard constraints. They posses no preassigned weights, meaning the planners are free to weigh them at their own discretion.

Preferences.

The problem can furthermore be extended by accounting for employee preferences, such as preferred colleagues and preferred days working/off, and other store preferences, such as a minimal occupation. It differs per supermarket what they are and whether they are implemented.

3. SHIFT SCHEDULING AS A CLASSIFICATION PROBLEM

Various solution techniques have been used for the shift scheduling problem. Most research applies a form of integer or mixed integer programming combined with linear or non-linear programming, or a constructive or improvement heuristic [7]. Meta-heuristics have been proven to be very efficient and produce effective solutions that are generally applicable [5], yet they are designed for their solving speed, possibly trading in parts of their optimality and accuracy. To be able to apply an ML approach, the shift scheduling problem is defined as a classification problem through descriptions of the input, the output and applicable evaluation metrics. The core idea behind ML is that the relationships between the different input variables leading to the desired output are initially unknown, but defined using realised examples from real-world data. This does not mean the constraints from section 2.2 are ignored; The constraints are examples of relationships that are discovered by the algorithm. An ML algorithm defines, trains and tests a model on the data, and uses the discovered relationships to predict the desired output on a new instance.

3.1 Input

The feature vector for shift scheduling in supermarkets includes, but is not limited to, the following employee and store features.

Employee.

- Skills
- Recurring rosters: [Bool, Bool, ..., Bool]
- Availability: [Bool, Bool, ..., Bool]
- Contract hours: float
- Labour costs: float
- Age: int
- Years of service: float
- Efficiency: float

Store.

- Expected Workload: [Bool, Bool, ..., Bool]
- Target Labour costs: [float, float, ..., float]

The listed variables were found in the researched literature and in the provided data-sets. The list is not exhaustive, and the feature vector can be extended to include other features.

As discussed, the skills of the employees correspond to the departments. In reality, rosters are initially constructed per department (planners are free to schedule departments concurrently), even if a single employee is scheduled to multiple departments. By having the ML algorithm simulate this, skill assignment is implemented, without needing to include them in the feature vector themselves. The time interval can be increased to reduce computation time and the probability of overfitting, since it significantly reduces the vector size. The recurring schedules and availabilities are formulated as Boolean lists, indicating for each time interval whether the employee is scheduled/available. Since ML algorithms generally take numbers as input, the booleans are represented by '1's and '0's. Age, years of service, and especially efficiency are features that are less actively used by planners, but is expected to still provide significant knowledge, since they were shown on the planning screen during the creation of the schedules in the data-set. Features like schedules from previous weeks, application of surcharges, leave hours (sick leave, holiday leave and time of work), etc., are deliberately omitted. This is either or both because of a high implementation complexity, or an expected lack of knowledge they will provide, considering their dependence on other features.

For each instance, the feature vector is a concatenation of the implemented features. This ensures that all feature vectors in a data set are of the same length. An example vector which implements all listed features is shown in Figure 2. The store's working hours range from 06:00 until 23:00, and a time interval of 1 hour are is applied, leading to 17 time intervals. The employee has no recurring schedule, and is available from 18:00 until 22:00. They have 10 contract hours, have a labour cost of 7,87, is 17 years old, have worked there for 1.58 years, and are 100% efficient.

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
 10.0, 7.87, 17, 1.58, 100,
 0, 0, 0, 0, 1, 2, 4, 6, 6, 4, 4, 4, 2, 2, 1, 1, 0
 0, 0, 0, 0, 34, 68, 136, 204, 204, 136, 136, 136, 68, 68, 34, 34, 0]
```

Figure 2. Feature vector example

3.2 Output

A prerequisite in more general personnel scheduling studies that also apply ML techniques, is that a day consists of shifts with predetermined starting times and lengths, which form the classes of the classifier. Since supermarkets manage variable starting times and lengths, the classes are adapted to follow the same format as the employee availabilities and recurring schedules, resulting in a Boolean list indicating for each time interval whether the employee under consideration is scheduled. Thus, each instance is assigned a label-set of the same size as the total number of classes, leading to the problem being multi-class and multi-label. An example label-set is shown in Figure 3, which corresponds to an employee scheduled from 19:00 until 22:00.

3.3 Model Evaluation Metrics

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0]
```

Figure 3. Label-set example

The classification accuracy of the model is evaluated using label-based metrics as described in [2]. Moreover, to evaluate how well the model solves the initial optimization problem, the predicted schedules are analyzed to determine how well they follow the sought-after constraints. An algorithm is constructed to do this analysis, since manually doing so is too time consuming. Additionally, usual benchmarks like basic tabu search or simulated annealing algorithms [5] could be implemented to further compare results, but these would also have to be adapted to apply to the multi-label problem. Therefore, the implemented metrics are compared to the available relevant literature.

3.3.1 Label-based metrics

For the evaluation of the classifier, the binary classes are initially considered independently. A confusion matrix is constructed per class where for each classification, it keeps track whether it was positive or negative, and whether the model had classified it correctly or incorrectly. Common binary evaluation metrics include recall and precision, as well as the F_1-value.

$$R = \frac{tp}{tp + fn}, \tag{1}$$

$$P = \frac{tp}{tp + fp}, \tag{2}$$

$$F_1 = \frac{2RP}{R + P} \tag{3}$$

Recall (Eq. 1) evaluates what percentage of positive instances are correctly classified as so. Precision (Eq. 2) evaluates what percentage of positively classified instances are correct. The mean of these two metrics is measured in the F_1-value (Eq. 3). 'tp' refers to all instances correctly classified as a positive, 'np' refers to all instances correctly classified as a positive, 'fp' to all instances incorrectly classified as positive, and 'fn' to all instances incorrectly classified as negative.

Shift scheduling is a multi-label problem, leading to multiple confusion matrices of different sizes being constructed. Because the importance of common labels might be higher, deciding the method of averaging the evaluation values influences the outcome. Two common averaging methods are the macro-average and micro-average.

$$M_{macro} = \frac{1}{q}\sum_{i=1}^{q} M(tp_i, fp_i, fn_i) \tag{4}$$

$$M_{micro} = M(\sum_{i=1}^{q} tp_i, \sum_{i=1}^{q} fp_i, \sum_{i=1}^{q} fn_i) \tag{5}$$

In the macro approach, the metrics are determined per class and then averaged, meaning equal weights are attributed to each class. The micro-approach first adds the values of the matrices together, creating a new confusion matrix, after which it determines the metric collectively. This approach applies equal weights to every single classification, placing a higher value on more common classes. 'q' refers to the total amount of labels.

3.3.2 Constraint-based metrics

Besides evaluating the classification accuracy of the applied ML algorithm, the integration of the defined constraints are measured by an algorithm of linear complexity. The constraints either apply to a single instance/employee, or to a single day. Per constraint, the fraction of instances that correctly followed the hard constraints is calculated using the following equation:

$$C_{hard} = c/t \tag{6}$$

'c' refers to the amount of instances/days that correctly introduced the hard constraints, and 't' to the total instances/days.

The introduced soft constraints are label-specific and apply to a single day. The accuracy of optimizing rosters according to the steering values is calculated using the following equation:

$$C_{soft} = 1 - \frac{|y - i|}{i} \tag{7}$$

'i' refers to the steering value, and 'y' to its realisation.

3.3.3 Comparison-based metrics

The constraint-based metrics are furthermore compared against those found in other literature on the scheduling problem in supermarkets, which was a single article at the time of writing [3]. Future research can use this paper as an initial benchmark for comparison of the label-based metrics,

4. METHODOLOGY

4.1 Data set

This research is carried out in association with the software company RetailSolutions [1], owners of the planning tool PMT (Personal Management Tool). RS has provided an SQL-dump from the database of one of their supermarket chains. Naturally, sensitive information was encoded in advance. MySQLWorkBench [2] was used for storing and processing of the data.

4.2 Input and Constraints

Due to time constraints, the feature vector described in section 3.1 is shortened to only include the employee properties 'Skills' and 'Availability', and only the 'Expected Workload' is included as steering value. A time interval of one hour is implemented, and the rosters from the data set have a maximum of one shift per day, where each shift is between one and eight hours.

4.3 Approach

The research of this paper was performed using a Neural Network (NN) model. An NN consists of an input layer, hidden layers, and an output layer, where each layer consists of neurons. Neurons of consecutive layers have many connections between them, each having a weight defining their importance. The NN passes the input through the hidden layers to the neurons of the output layer, a process which mimics the human brain. After initializing the model, its weights and other variables are adjusted to examine their effect on the predictions.

[1]https://www.retailsolutions.nl/

[2]https://www.mysql.com/products/workbench/

4.3.1 Preparing the data

A Python script was constructed and applied to convert the raw data into a usable .csv file, ensuring the features have the desired format presented in section 3.1. Each file consists of N rows, and each row represents the input and output pertaining to an employee from department z_1 in store z_2 on day z_3. The MySQL library was used to connect to the MySQL database and execute queries on the data set.

First, the relevant employees are extracted by selecting employees who had specified any availability on the indicated day. This is followed by the construction of the employee's feature vectors, created per feature. The extracted employee's availability is formulated as a list of time segments, indicating for each segment the starting and ending time, as well as the type (f.e. School, Sport, Available). This list is converted into the desired binary list format, stating for each time interval whether the employee is available or not. The extracted expected workload is formulated as a list of processes, stating for each process its starting time, ending time, and workload in minutes. The expected workload per time interval is computed and represented as the desired floating point list. The employees' realised rosters are extracted and converted into the desired binary list, similar to the availabilities. Finally, these values are concatenated to form the rows of the data set. Each row includes the employee's availability, the store's expected workload, and the employee's roster. An example of an employee who is available from 18:00 until 22:00 and scheduled from 19:00 until 22:00 is shown in Figure 4, on a day where the expected workload represents a skewed normal distribution.

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
 0, 0, 0, 0, 1, 2, 4, 6, 6, 4, 4, 4, 2, 2, 1, 1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0]
```

Figure 4. Data row example

4.3.2 Defining the model

The PyTorch library is used to define, train and evaluate the model. An NN is initialized, by extending the PyTorch Module class, a. It is defined by layers, their weights and activation function, and a forward function. The input layer consists of 34 neurons, corresponding to the size of the feature vector. The output layer has 17 neurons. The model has 2 hidden layers of which the amount of neurons is evenly distributed between the input and output layers, resulting in the first layer having 28 and the second layer having 23. Both hidden layers, as well as the output layer, are linear. This means every connection between the neurons performs a linear transformation (Eq. 8) on the passing data.

$$y = wx + b \tag{8}$$

'w' refers to the weight of the transformation, and 'b' to the bias. Xavier initialization is used to assign the initial weights, distributing them uniformly between two values dependant on the layers' input and output sizes. The output layer is assigned a Sigmoid activation function, converting the results of the transformation to a value in the range (0, 1). A forward function is constructed to guide the data through the layers and the appropriate activation function. The constructed model is illustrated in Figure 5.

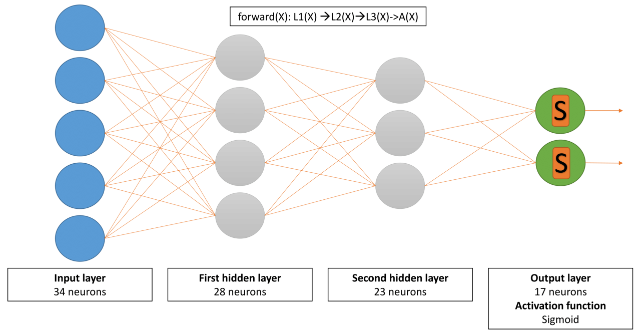

Figure 5. The constructed neural network

4.3.3 Training the model

The Pandas library was used to read the .csv file. The rows are randomly split into a train set and a test set, following a 2:1 ratio. The training of the model is performed by iterating over a loss function and an optimization algorithm. The loss between the predicted output and the target output is computed using the cross entropy loss function. A Stochastic Gradient Descent (SGD) algorithm is implemented as the optimization algorithm. The SGD takes a learning rate lr and a momentum m as variables, which are initialized as lr = 0.1 and m = 0.9, but adjusted to improve performance. The learning rate defines the impact the optimization each iteration has on the weights. The momentum further optimizes the SGD, by directly influencing each weight adjustment. The model runs for 1000 iterations. Each iteration starts by passing the input through the model. The loss function is applied to calculate the loss of the generated output. The SGD computes the gradient of the loss function at the determined point, and updates the weights for each layer using both the gradient and the momentum, aiming to reduce the loss of the next iteration.

4.3.4 Evaluating the model

The evaluation of the model is performed using the evaluation metrics described in section 3.3. The trained model is used to predict the rosters of the employees in the test set. These predictions, together with the corresponding target rosters, are then used to compute the metrics.

The macro-average and micro-average of the F_1-values are implemented through the Sci-Kit Learn library.

Honouring employee skills and time intervals are is inherently achieved by treating shift scheduling problem in supermarkets as a classification problem, as described in section 3.1. Another Python algorithm is constructed to examine how well the constructed rosters honour employee availabilities, a maximum of one shift per day, and a maximum of eight hours per shift, using Eq. 6. It furthermore calculates the workload coverage using Eq. 7, where the final workload coverage is the average of all the intervals.

The aforementioned article [3] uses the same workload coverage formula as presented in equation 7, granted that they average them over all the departments of a store. Their results show the workload coverage of 50 stores using their solution, for which the number of employees ranges from 78 to 772. Each store (except for one) has a value between 70% and 80%.

4.4 Improving accuracy

The data preparation, model construction and model training are all of high adaptability. Because of time constraints, three variables that are expected to have a major influence on the model accuracy are researched. An appropriate ratio between the train set and test set size is determined in advance by testing the accuracy of a 1:2, 1:3 and 1:4 ratio.

The first researched variable is the data size N, which is directly dependant on the selected department z_1, store z_2 and day z_3. The accuracy is measured for N ranging from 5 to 35, with approximate increments of 5. These N are found by manually searching the data.

The influence of the learning rate lr is measured by starting at 0.1, and logarithmicly decreasing it to 0.0001. The number of iterations e is increased, such e * lr = 100.

Similarly, the momentum m is analyzed, by having m range from 0.1 to 0.9, with increments of 0.2.

5. RESULTS

Each individual test was run 100 times, after which the average values were noted in this section's tables.

5.1 Train and test set ratio

With N, lr, and m set at 22, 0.1, and 0.9, respectively, the results of changing the train to test set ratios 1:2, 1:3, and 1:4 are shown in Table 1. Since there is no apparent distinction in between the evaluations of the different ratios, the ratio 1:2 was used in further tests.

Table 1. Adjusting the train and test set ratio r

r	Macro F_1	Micro F_1	$C_{availability}$	$C_{maxshift}$	$C_{maxhours}$	$C_{workload}$
1:2	0.652	0.748	0.817	0.056	0.999	**0.174**
1:3	**0.660**	**0.750**	**0.852**	0.048	**1.000**	0.048
1:4	0.647	0.744	0.805	**0.068**	0.993	0.113

5.2 Data size

The results of adjusting the input size N are shown Table 2. Increasing the size of the training set generally improves the accuracy of resulting model, which is confirmed by the increase in accuracy of label-based metrics as N increases. Increasing the data size furthermore leads to a worse honouring of the employee's availability and a maximum of one shift. This is likely because a smaller N leads to less room for mistakes. The honouring of a single shift is also less respected when N increases. This could be because the employees generally have long scheduled schedules. This would also explain less honouring of the maximum of 8 hours, since long shifts may exceed this boundary. A data size of 20 shows the best overall results.

Table 2. Adjusting the data set size N

N	Macro F_1	Micro F_1	$C_{availability}$	$C_{maxshift}$	$C_{maxhours}$	$C_{workload}$
6	0.505	0.567	**1.000**	**0.880**	0.650	0.186
13	0.656	0.740	0.865	0.480	0.970	0.173
15	0.680	0.781	0.826	0.034	**1.000**	0.175
20	**0.744**	**0.832**	0.856	0.103	**1.000**	0.174
24	0.688	0.783	0.784	0.087	0.998	**0.264**
30	0.668	0.732	0.752	0.298	0.984	0.231

5.3 Learning rate

Learning rates from 0.1 decreasing to 0.00001 have been tested, as seen in Table 3. It appears that the label-based metrics decrease when the learning rate is set below 0.001, yet show no significant change for values above that. The same is true for honouring availability, honouring the maximum shifts per day and the workload coverage, although a high learning rate also seems to imply lower values. The honouring of maximum hours per shift is unaffected by the learning rate, but this is most likely because employees are scheduled over nonconsecutive time intervals, which is supported by the low honouring of maximum shifts. A learning rate of 0.01 shows the best overall results.

Table 3. Adjusting the learning rate lr

lr	Macro F_1	Micro F_1	$C_{availability}$	$C_{maxshift}$	$C_{maxhours}$	$C_{workload}$
0.1	0.715	0.803	0.771	0.040	0.990	**0.174**
0.01	**0.744**	0.832	**0.856**	**0.103**	1.000	**0.174**
0.001	**0.744**	**0.835**	0.847	0.069	1.000	0.170
0.0001	0.697	0.788	0.749	0.074	1.000	0.171
0.00001	0.579	0.668	0.586	0.030	1.000	0.163

5.4 Momentum

Momenti between 0.1 and 0.9 have been tested, of which the results are shown in Table 4. A higher momentum seems to implicate a worse label-based evaluation, though it is minimal. However, minimizing this value also has a negative impact. The honouring of the availability decreases when the momentum increases, while the honouring of maximum shifts per day, the honouring of maximum hours per shift, and the workload coverage stay approximately equal. A momentum of 0.3 shows the best overall results.

Table 4. Adjusting the momentum m

m	Macro F_1	Micro F_1	$C_{availability}$	$C_{maxshift}$	$C_{maxhours}$	$C_{workload}$
0.1	0.750	0.838	**0.870**	0.056	**1.000**	0.172
0.3	**0.753**	**0.841**	0.864	0.026	**1.000**	0.171
0.5	0.750	0.837	0.849	0.046	**1.000**	0.172
0.7	0.743	0.830	0.849	**0.063**	**1.000**	0.172
0.9	0.715	0.803	0.771	0.040	0.990	**0.174**

5.5 Optimal solution

When the optimal N, lr and m found in the previous paragraphs, we get the results found in Table 5. The metrics are indeed high compared to the best scores of the previous results. The label-based metrics are not optimal, but show that the model has a partial understanding of the relation between the input and the output. Considering the algorithm is not specifically instructed to honour the availability, 0.8 can be considered accurate. However, the other constraints are not followed.

Table 5. Applying optimal values

N	lr	m	Macro F_1	Micro F_1	$C_{availability}$	$C_{maxshift}$	$C_{maxhours}$	$C_{workload}$
20	0.01	0.3	0.763	0.824	0.804	0.033	1.000	0.169

Using this model, a single prediction is made on a single feature vector, randomly selected from the test set. The feature vector, predicted roster and actual roster are shown in Figure 6, 7 and 8, respectively. The model has scheduled the employee honouring their availability, and maximum amount of hours per shift. Although, it has incorrectly distributed the hours over nonconsecutive time intervals. It furthermore shows that the prediction has scheduled the employee significantly more than they actually were.

5.6 Analysis

The label-based metrics illustrate that the models have an understanding of the problem, purely based on its predictions. However, apart from the availability constraint, the constructed NN does not sufficiently follow the predefined constraints. The other research on shift scheduling in supermarkets [3] has achieved a significantly higher workload coverage then the introduced NN. There are multiple explanations for this.

```
[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
0.38235294818878174, 0.39901959896087646, 0.7740195989608765,
1.0240195989608765, 0.39901959896087646, 0.39901959896087646,
0.5740196108818054, 0.740686297416687, 1.1573529243469238,
1.1573529243469238, 1.1990195512771606, 9.632352828979492,
9.632352828979492, 8.549019813537598, 9.424019813537598,
9.424019813537598, 0.38235294818878174]
```

Figure 6. Feature vector example

```
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0]
```

Figure 7. Prediction example

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0]
```

Figure 8. Actual example

Firstly, the constructed data sets are restricted to individual days, leading to a small N. Previous results have shown that increasing N generally leads to better results, which is generally true with an NN. The data sets of consecutive days could be concatenated, assuming that they were constructed by the same planner.

Secondly, the number of implemented features is very low compared to those available. The information from the excluded features is lost, which a definitely a large part of the accuracy loss.

Thirdly, the applied time interval of one hour is more than the 15-minute interval used for the actual schedules. Applying the latter will allow for a more precise planning, yet adds a remarkable amount of features, which makes the problem a lot more complex.

Fourthly, the initialization of the NN has multiple variables, which could be adjusted to examine their impact on the results. The layer sizes of the model are evenly distributed between the input and output size. However, no significant research has been done into finding the optimal sizes. Moreover, the weights of the connections are initialized using Xavier initialization. This has only been compared against the He weight method, and could still be improved. Furthermore, the number of iterations in the training stage of the model is set 1000, which has shown more accurate results than lower values, but no significant improvements a higher values.

6. DISCUSSION

It is obvious that the demonstrated results are far from optimal. Still, the introduced metrics are high enough to conclude that even the simplest forms of NN are able to provide some accurate results. Other ML algorithms could also be introduced to help solve this problem, because different approaches will excel in different evaluation metrics. Since different algorithms use different techniques, they could each help improve accuracy in their own way. Moreover, extending the model with more multi-label classifiers, and creating problem specific ensembling techniques, further helps overcome problems like label imbalance, as shown in [4]. Comparing the achieved results to other literature is difficult, apart from the analysis in section 5.6, since the employed problem definition is unconventional, and many researches on the general scheduling problem merely provide theoretical solutions. The best way to continue the research would be by improving the NN as described in section 5.6, since the problem should first be solved more accurately for it to have any actual use. Treating the complex shift scheduling problem in supermarkets as a classification problem, and applying a machine learning algorithm to solve it, has shown promising initial results, t hough it can not yet compete with state of the art search algorithms. If accurate results have been achieved, the shift scheduling classification problem can be extended to tour scheduling and task assignment as described in the introduction, and solved using this constructed NN.

7. CONCLUSIONS

This research has applied a neural network approach to to multi-label shift scheduling problem in supermarkets. The shift scheduling problem in supermarkets is defined by a set of hard and soft constraints pertaining to the rosters of employees. This problem is converted into a multi-label classification problem by treating employees as feature vectors, each conveying information relevant for scheduling. The developed neural network is able to generate rosters based on these vectors, though be it with low accuracy. The network itself can be improved in many ways, and many more features can be introduced to improve its accuracy. However, the results have shown that treating the shift scheduling problem in supermarkets as a classification problem, and using a neural network to solve it, is able to partially solve it.

8. REFERENCES

[1] A. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, Feb 2004.

[2] E. Gibaja and S. Ventura. A tutorial on multi-label learning. *ACM Computing Surveys*, 47, 04 2015.

[3] S. K. Mirrazavi and H. Beringer. A web-based workforce management system for sainsburys supermarkets ltd. *Annals of Operations Research*, 155(1):437–457, 2007.

[4] M. A. Tahir, J. Kittler, and A. Bouridane. Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recognition Letters*, 33(5):513–523, 2012.

[5] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

[6] S. Zolfaghari, V. Quan, A. E. Bouri, and M. Khashayardoust. Application of a genetic algorithm to staff scheduling in retail sector. *International Journal of Industrial and Systems Engineering*, 5(1):20, 2010.

[7] E. H. Özder, E. Özcan, and T. Eren. A systematic literature review for personnel scheduling problems. *International Journal of Information Technology amp; Decision Making*, 19(06):1695–1735, 2020.