Predictive planning with neural networks

Russell Ainslie, John McCall
School of Computing Science and Digital Media
Robert Gordon University, Aberdeen, UK

Sid Shakya, Gilbert Owusu
Business Modelling and Operational Transformation Practice
British Telecom, Adastral Park, Ipswich, UK

***Abstract*—Critical for successful operations of service industries, such as telecoms, utility companies and logistic companies, is the service chain planning process. This involves optimizing resources against expected demand to maximize the utilization and minimize the wastage, which in turn maximizes revenue whilst minimizing the cost. This is increasingly involving the automation of the planning process. However, due to unforeseen factors, the calculated optimal allocation of resources to complete tasks often does not match up with what is actually occurring on the day. This factor highlights a requirement for a method of predicting accurately the number of tasks that will be completed given a known amount of resources and demand in order to produce a more accurate plan.**

In this paper we define a predictive planning problem and present an application of Neural Networks (NN) to solve this problem. We first investigate the best setup for this NN based model to solve the general problem, producing a very high accuracy whilst ruling out the need for the addition of any additional inputs or techniques that would have additional computational and storage requirements.

Finally we refine this model further for a specific real world scenario to a point where we prove it provides superior accuracy to the current planning procedures. This fulfils the requirements of the predictive planning problem and thus provides a model that can be used to support automated planning optimization.

Keywords— NN; Neural Network; Prediction; Tactical Planning

I. Introduction

In recent decades, with the advent of privatization and the increase in competition, there has been a drive in the service operations sector to improve the sophistication of supporting applications up to the level of the far more mature supply sector [1]. Recent research has shown that this is increasingly involving the automation of planning processes [2].

One of the key areas for development lies within the field of service chain planning. Planning for a service industry involves the matching of resources' available time to the jobs requiring completion. This matching occurs along the dimensions of time, area and skill. Each job requires a skill to complete, in a certain geographical area, within a certain period of the plan. The planner attempts to match this with the skills and areas a resource can cover within each period, by using some simple rules [3]. The key objective of planning is to have the right resource available at the right time in the right place to fulfill the customer demand. Advanced planning of resources helps firms to maximize the utilization and minimize the wastage, and by doing so it helps to fulfill customer demand and maximize revenue, at the same time minimizing cost. Planning can be categorized into three groups [4].

1. Strategic planning - a long term plan looking 3-5 years ahead deciding company strategy for the next few years. [5]
2. Operational planning - a medium term plan looking at 12 to 18 months ahead taking operational decisions such as demand prioritization and resource recruitment. [6]
3. Tactical planning– a short term plan up to 90 days ahead balancing daily capacity against daily workstack [7].

We will be focusing mainly on tactical planning in the rest of the paper. Tactical planning is typically done on a daily basis up to 90 days in advance. However much of the input from the planner is required for the first 7 to 14 days, i.e. near the execution date, where the unforeseen fluctuation in available resource capacity as well as change in demand has to be taken into account to produce an actionable plan to match demand to supply. Depending upon the type of task to be undertaken, the planning process can be different. For example, tasks that are related to faults and require repair can be predicted in advance due to their repetitive nature and past patterns of fault volumes. Depending upon the company's preference, resources can be allocated to complete these expected fault related tasks in advance. Furthermore, the number of completions can also be predicted in advance based on the expected tasks and the available capacity for that day. The completions can be different to the actual tasks due to a number of factors, such as failure rate, cancellations, spare parts not being ready etc. It is important to predict the completion to accurately know how much resource will be available to allocate to other types of tasks. This also gives planner's first signal on what to allocate for faults. These predictions are particularly helpful for future periods where their figures can be automatically populated as the starting completions, removing the need for planners to invest time to review the full 90 days. They only need to make changes to these initial figures if there is a deviation in capacity or demand from the original prediction. This saves planner's time as they do not need to plan for 90 days and also can make the plan more accurate as historical completion patterns are taken into account when making planning decisions, not over or underestimating completions. In this paper we investigate methods to accurately predict completions for tactical planning purposes.

The paper is organized as below. Section 2 describes the predictive planning problem and defines the models that are used in predictive planning with justification to use a neural network (NN) [8] [9] [10]. Section 3 introduces three different predictive planning models investigated in this paper. It also describes the data set and the experimental setups. Sections 4, 5 and 6 describe the three models respectively together with the experimental results. We conclude the paper in section 7.

II. The predictive Planning Problem

This section of the paper first outlines the predictive planning problem before possible predictive methods to solve this problem are described along with the justification for choosing the NN which we go on to develop in the remaining sections of this paper.

A. Problem Definition

The predictive planning problem is that of predicting the expected number of completions of each task type (tasks grouped by the skill required to complete them) by each resource type on each day of the plan given the available time of each resource type (the capacity) on each day, the number of tasks requiring completion for each skill at the start of the plan (the workstacks) and the expected number of new tasks arriving (the intake) for each skill on each day.

For the purpose of this paper, the problem can be simplified to that of a single day. The requirement to predict the completions that day, given the current workstack levels and the capacity. The predicted completions on this day can then be used to calculate the expected workstacks for the next day by subtracting them from today's workstacks and adding on the intake. This would fulfil the requirements of the full definition, however some prediction errors would propagate to future days of the plan. This is not a problem however as the plan is updated each day, thus the current day would not contain any additional errors and the future days values are merely used to forecast any potential problems that may arise and set some base expectations for what is likely to occur.

A separate model will be required to predict completions for each skill by each resource as knowledge of who is completing what is required in the final produced plan.

B. Predictive Methods

Methods that were considered for solving this problem were the use of a rolling average, which is simply taking the average of the past x number of day's completions for that skill by that resource type on that day of the week, a linear regression [11] or a NN.

The rolling average was only briefly considered, although its accuracy was investigated, as it would not take into account the varying capacity levels on a given day. For example, if resourcing levels were lower than previous weeks the rolling average would be predicting completion levels that were impossible to meet with that capacity. Clearly this is not good enough for a predictive planning model. However, the rolling average did make for a good baseline comparison value to use during the early stages of later model development.

The second method considered was the use of a linear regression. A linear regression is an approach for modelling the relationship between an output variable and one or more input variables. This could be used to produce an equation to be used to calculate the output variables given our known inputs. This would take into account the capacity levels if they were used as an input to the model and thus is an improvement over the rolling average. Some initial testing however showed the linear regression approach was not providing much, if any, improvement over the rolling average. This is likely due to some complex relationships between variables that the regression was unable to capture.

Thus, the third method was considered, that of using a NN. For the purposes of this paper we are focusing on the multi-layer perceptron (MLP) [12] variant of NN as they are widely used in forecasting problems [13].

MLP's incorporate three layers, an input layer, an output layer, and between those, the hidden layer. The hidden layer can contain a number of layers within itself. Each layer can contain a number of nodes, each of these nodes is connected to each of the nodes in the next layer. The number of layers within the hidden layer and the number of nodes in each of these layers will be referred to as the hidden layer topology for the remainder of this paper. The number of nodes in the input layer will always equal the number of inputs to the model, and the number of output nodes will always be one. This is because we will be building a separate model to predict each of the required output values. Fig. 1 shows an example of the layout of an MLP with three inputs and a hidden layer topology of two layers each with two nodes (or 2-2). The connections between the nodes in each layer are each assigned a weight, these combine with node biases to allow the network to perform complex non-linear calculations giving an output value based on the values input to the network. This non-linear feature is the last element required by our predictive methods to capture all the elements of our modelling problem. Thus the NN, in particular the MLP, is a suitable candidate for use as the core model of the application and is the focus of the remainder of this paper.

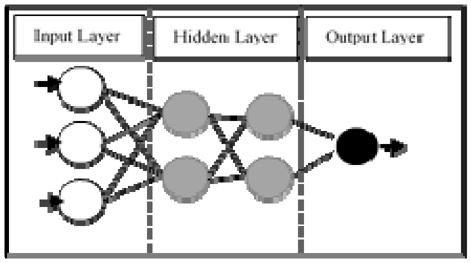

Fig. 1. Multi-layer perceptron node layout example

III. Neural Network Model Development

During the model development process a large number of experiments were performed to evaluate and validate decisions made. All of the results presented are generated using real data from a member of the service industry with a large multi-skilled workforce. The data obtained contained three workforce types, six fault skills and seven installation skills. The specific data gathered for the model were the daily values for the capacity of the three workforce types and the total workstack levels of the thirteen skills, used as inputs, along the number of tasks completed by each resource type, used as the output of each model. This data was gathered for four separate areas. Two separate historical data sets were available for model building purposes, 1) 38 weeks data for a specific day of the week - Friday. 2) 6 weeks data for all days of the week.

Due to this data constraint, we developed three separate NN models to cover three different scenarios.

1. Single day model – to evaluate the full accuracy of a day specific completion prediction model with a large dataset of 38 weeks
2. Full week model – to extend the single day model to predict completions for the whole week with a limited set of data
3. Fault only model – Specific completion prediction model for some products where installation data is not relevant and only fault data is available

For each experiment in this paper, a separate model was created to predict the completions of each of the six fault tasks by each of the three workforce types in each of the four areas, resulting in a total of 72 models per experiment. These models were implemented using Java and the Encog NN library [14] and run on a 64 bit Windows 8.1 based machine comprising 4GB RAM and an Intel Core i5-4300U CPU. One week's worth of data was set aside as the test data set and the models were trained on the remainder. The trained models were then run on the test data and the predicted values were compared with the actuals to calculate the accuracy and Pearson's correlation for each task completion in each area. The average of the accuracies and correlations over the four areas are what is presented and analyzed in this paper.

IV. Single day model

For the single day model, initially a MLP was implemented with two hidden layers, both containing five nodes, using the backpropagation [15] training algorithm. All nodes were set to use the sigmoid activation function [16]. Training was set to run for 500 iterations. We then tuned this further through two distinct phases.

1. Input analysis: Analysis and decisions relating to the inclusion of possible inputs.
2. Training algorithms comparison

A. *Phase 1: Input Analysis*

The first decision to be made with regards to the model inputs was how, if at all, to use the past completion data as an additional input. Two alternatives were tested, inputting these as a time series or as a rolling average. In both cases the number of past data points included was varied from zero (no past completion data used) up to eleven. The accuracy achieved for each of these values can be seen in Table 1. It was discovered that the accuracy was similar between the two with the rolling average slightly more accurate. The decision was made to use the rolling average as the past completion input, with a duration of three selected. This gave the best correlation value and also kept the average length low which kept the number of lost data points to a minimum whilst still giving a slight accuracy increase over not using the average at all.

The second decision was which of the remaining inputs to use. A correlation analysis between the inputs and outputs was performed, some typical results shown in Fig. 2 and 3. The graphs show the correlation of the stated output with each of the inputs, the capacity for resource 1 to resource 3 and the workstacks of skill 1 to skill 13. Here skills 1 to 6 were the fault skills and 7 to 13 were the installation skills. Fig. 2 shows an example of typical correlation results seen across a large number of outputs. The output of completions of skill 2 by resource 1 are most highly correlated to the workstack for that skill (as you would expect, the more work available the more jobs that you may complete). However between the four areas we can see that the correlation with the remaining inputs varies significantly. Fig. 3 shows a less typical situation where we see that completions of skill 4 by resource 2 are more highly correlated to some of the other skills workstacks than its own workstack. This could be due to this resource only working on tasks of this skill type if there are not enough of other types available or perhaps this task is often performed at the same time as another. Again we see a large variance between correlations with different inputs between the areas, thus the decision of which inputs to use could not just be made on an output by output basis. In order to accommodate this, and also any possible future correlation differences, a dynamic method of choosing which inputs to use was created. During training, the correlation between each input and the output was calculated and inputs were filtered out if they did not pass a set threshold.

This threshold was the focus of the experiments displayed in Table 2. The first experiment run was using a length 3 rolling average and all of the capacity and demand inputs. This showed an improvement was achieved by dynamically filtering out any inputs below a certain correlation value, with a cut-off point of 0.3 achieving the best results.

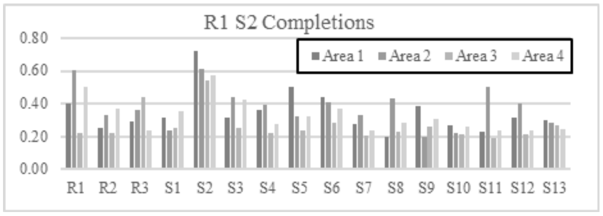

Fig. 2. Typical correlation results for the model outputs

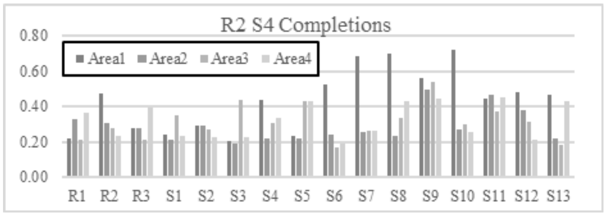

Fig. 3. Case where the output is not most correlated with its workstack

Thus at the end of phase one we can conclude that a rolling average of length 3 and a correlation cut-off point of 0.3 provides the best results. However both of these processes do have their cons. In the rolling average case the average requires additional data and calculations for each day, also once you are talking about future dates would include predicted values introducing some errors. For the correlation, the calculations involved introduce additional computation during the training process. It also introduces additional storage requirements when saving each model, as which inputs were used in each case would have to be stored along with the model itself. For these reasons, the gains achieved through including these factors continues to be evaluated through the additional developments in phases two and three to ensure they still outweigh the cons.

TABLE I. PAST COMPLETIONS AS INPUTS

Data Points	**Average**		**Time Series**	
	Accuracy	***Correlation***	***Accuracy***	***Correlation***
0	83.8%	0.54	83.8%	0.54
1	83.5%	0.54	83.5%	0.54
3	84.4%	0.56	84.2%	0.53
5	84.7%	0.53	84.3%	0.53
7	84.8%	0.51	84.4%	0.52
9	85.1%	0.48	84.5%	0.49
11	85.0%	0.50	84.5%	0.47

TABLE II. CORRELATION FILTERING

Correlation Filter	**Backpropagation**		**Resilient Backprop**	
	Accuracy	***Correlation***	***Accuracy***	***Correlation***
0	84.4%	0.56	88.3%	0.73
0.1	84.3%	0.54	88.2%	0.72
0.3	85.4%	0.61	88.9%	0.77
0.35	85.2%	0.61	88.6%	0.74
0.4	85.0%	0.60	88.7%	0.71
0.45	85.2%	0.60	87.6%	0.70
0.5	84.9%	0.60	86.6%	0.68

B. *Phase 2: Training Algorithm comparison*

The next step undertaken was to explore some alternate training algorithms available within the Encog library. Resilient backpropagation [17] was selected as a good candidate and tested, the results shown in the second column in Table 2. It was found to produce far superior results to the original backpropagation. In particular it showed a large improvement in the correlation. Thus it was used throughout the future development. It was also noticed that with the improved training algorithm the improvements from using correlation filtering had been reduced. It was decided to remove the correlation filtering step to simplify the training and storage processes as it wasn't providing a great enough improvement to justify the complication.

V. FULL WEEK MODEL

The next step was to expand the single day model through the creation of a model to cover an entire week. With only six weeks of data available there were not enough data points to create an individual model for each day of the week. As such three methods for constructing a model to cover the entire week were investigated.

1. Create one NN model to predict any day of the week.
2. Create a composite model combining one NN model to predict weekdays and another to predict weekends as weekends tended to behave differently to weekdays.
3. After observing that Saturdays often only had a slight reduction in completion values over a weekday a third alternative was investigated, to create the composite model using one NN model to predict Monday to Saturday combined with a separate Sunday model.

With the correlation filtering already ruled out during the single day model building process, we continued to evaluate the rolling average to ensure the gains were still enough to overcome the cons. For the full week data this average is constructed for each day of the week individually, so the rolling average input on a Monday would be the average of the last x Monday's completions. The results of testing the different model configurations, along with the varying number of data points used for the average, can be seen in Table 3.

These showed that the best combination was the third option, to create a NN model for Monday to Saturday with a separate model for Sundays. The inclusion of the average also produced little improvement in accuracy or correlation, with the model producing a very high correlation to the weekday trends (e.g. lower at the weekends) even without the average to give day specific inputs. In some cases the average even reduced the model accuracy. Thus it was decided the best decision was to not use the rolling average as an additional input.

VI. Fault Only Model

Unlike fault workstacks, which can be forecast many months in advance, in many products and services the installation workstack is only fully known on the execution day. This means that these workstacks are not available to use as inputs for these products. The rest of this paper is focused on solving this specific scenario as it is one of the core requirements from the planning community. The goal is to produce a prediction at least as accurate as the current manual planning process where planners decide the completions numbers only based on the forecasted fault workstack.

The first approach was to simply run the model after removing the installation data set. Initial tests showed the installation workstacks being unavailable for use as inputs caused a significant dip in the accuracy of the full week model of greater than 10%.

A two phase approach was then investigated to improve the prediction model to achieve better results than the current planning processes.

1. Initial Network Improvements: The first phase was an attempt to improve the neural network to produce a greater accuracy. For this we introduce cross validation into the training process, and also investigate different hidden layer topologies.
2. Overfitting avoidance: The second phase was to investigate solutions to the problem of overfitting to the training dataset. This was motivated when an initial test run of an experiment to change the number of training iterations noted an accuracy decrease when the number was initially increased. This highlighted that the accuracy is perhaps being impacted by overfitting to the currently small training data set.

We describe the two phases in detail below in sections A and B, followed by the final analysis of the results in section C.

TABLE III. Full Week Model Setups

Data Points	**All Week**		**Weekday + Weekend**		**Mon-Sat + Sun**	
	Acc.	***Corr.***	***Acc.***	***Corr.***	***Acc.***	***Corr.***
0	86.1%	0.93	87.2%	0.93	88.3%	0.94
3	85.7%	0.92	87.6%	0.94	88.0%	0.94
5	85.6%	0.92	87.8%	0.94	88.5%	0.95
7	85.8%	0.93	87.7%	0.94	88.2%	0.94
9	86.5%	0.93	87.8%	0.94	88.1%	0.94

A. Phase 1: Initial Network Improvements

As mentioned above, the first experiments were an attempt to increase the accuracy by improving the network. The first of these involved introducing cross validation to the training process. The number of folds were varied, with values ranging from two to six tested, to discover which made best use of the available training dataset. The best of these were used in the subsequent stages. The second round of experiments involved modifying the topology of the hidden layer. Early on it was discovered that using just one single hidden layer appeared to be optimal, with values from three to nine hidden nodes tested, however a few multi-layer results are also included.

1) Cross Validation Results

Fig. 4 shows the results of varying the number of folds when using cross validation. The difference is not very large, however the best result for accuracy and correlation can be seen around three folds where 76.1% is reached. At this stage the model accuracy is still below the planners' accuracy of 82.6%, however if we look at the accuracy breakdown per task type seen in Fig. 5 for the best result of three folds we can see that in the case of task types 2 and 3 the model is producing slightly better results. However the model is underperforming substantially when predicting the remaining task types.

However there is still a large deficit in the remaining two task types.

The results look promising overall however with such a significant improvement achieved.

2) Hidden Layer Topology Results

In Fig. 6 We can see the results of the hidden layer topology tests. A single number for the node topology indicates that number of nodes in a single layer, two numbers indicates that number of nodes in each of two layers. The results show that further improvements have been reached over the 5-5 topology used previously. The best results occur for using only a single hidden layer with the optimum occurring with 8 nodes in that layer producing an accuracy of 79%. This is still lower than the planners' accuracy but has halved the gap previously seen. The correlation has also bridged the gap by a similar magnitude. Looking at the task type breakdown for this improved model (Fig. 7) we see that task type 2's accuracy is further improved compared to the planner, with task types 1, 3, and 6 now reasonably close.

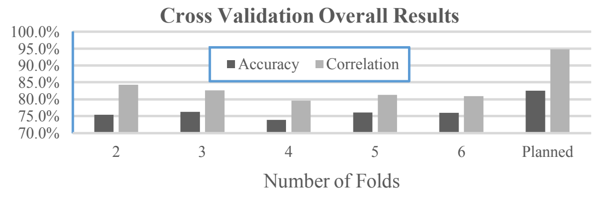

Fig. 4. Cross Validation Experiment Overall Results

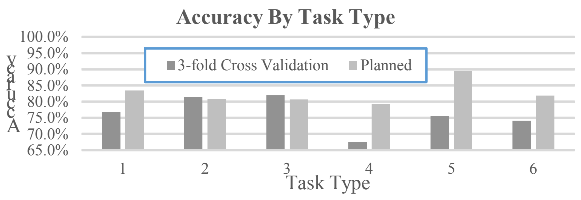

Fig. 5. 3-Fold Cross Validation Accuracy Results By Task Type

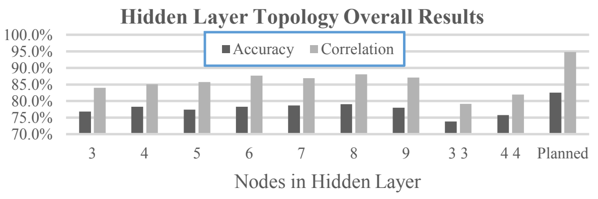

Fig. 6. Hidden Layer Topology Overall Experiment Results

B. *Phase 2: Overfitting avoidance*

The next batch of experiments performed for phase 2 involved attempting to overcome the overfitting to the small dataset. The first solution trialed was using a cluster of models, rather than a single model. Multiple models were trained for each output and the predicted value given by taking the average of the predicted value of all the models in that cluster. The final method explored to overcome the overfitting issue was the use of a new stopping condition during the training process. Instead of just running the training for x number of generations, something which we could have attempted to tune in the same manner as the hidden layer topology, it was instead decided to use a more dynamic stopping condition, that of the early stopping strategy [18]. During training, instead of using all of the training data set to train the model a further portion is set aside for validation at each training generation. After each training generation the accuracy is tested on the separate validation data, the theory being that as the model is trained the accuracy will initially increase for this validation data. However, once overfitting starts to occur the accuracy will then begin to decrease again on the validation data. Thus the early stopping strategy runs the training process until the accuracy on the validation data begins to decrease, e.g. once overfitting has been detected.

1) *Model Cluster Results*

This next experiment involved training a cluster of models to attempt to reduce the impact of overfitting by taking their average as the actual prediction. The results in Fig. 8 show that there is a further, slight, improvement achieved of around 1% with the best coming from using a batch of 7 models giving an accuracy of 80.1%. Looking at the individual task type breakdown for this setup in Fig. 9 we can see that it provides better results than the planner in half of them now, however there is still a large deficit in task types 4 and 5.

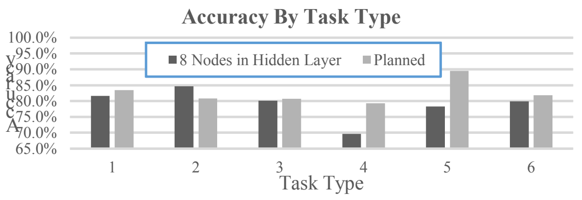

Fig. 7. 8 Nodes in Hidden Layer Accuracy By Task Type

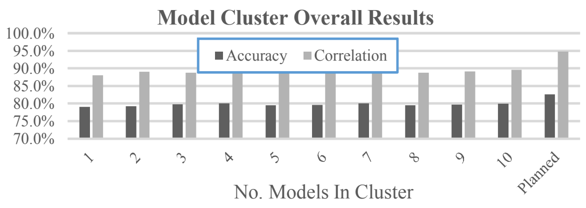

Fig. 8. Model Cluster Overall Experiment Results

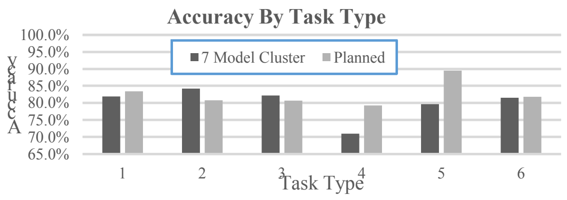

Fig. 9. 7 Models In Cluster Accuracy By Task Type

2) *Early Stopping Strategy Results*

The final experiment performed was a repetition of the previous experiment, varying the number of models in the prediction cluster, but with the addition of the use of the early stopping strategy during model training to avoid overfitting to the small data set. This produced significant improvements, as seen in Fig. 10, with the best overall accuracy now achieved with a 5 model cluster. This gave an accuracy of 84.1%, which is higher than the 82.6% achieved by the planners in the same period. In fact, even without using clustering (with a cluster size of 1) the model was achieving 83.1% which is still slightly better than the current planning process. The correlation achieved is also up to 0.94, which is almost the same as the 0.95 achieved by the planners. Looking at the individual task type breakdown again in Fig. 11 we can see that the model is now outperforming the planner in five of them, however there is still a significant deficit in task type 5. This may highlight this task type as one that is a problem for the model to predict accurately or the manual planners may understand something about it that has not been captured by the inputs being used.

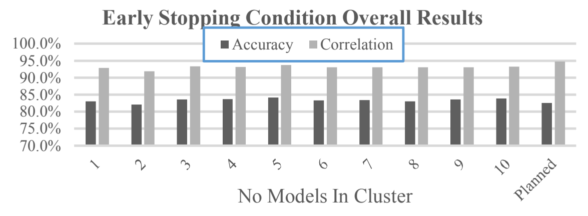

Fig. 10. Early Stopping Training Condition Overall Experiment Results

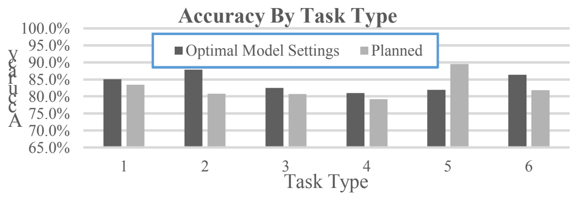

Fig. 11. Optimal Model Settings Accuracy By Task Type

C. Prediction Quality

In Fig. 12, 13 and 14 we look more in depth at how the optimally setup model is predicting the weekly trend. These graphs show the total actual completions per day across the four areas, against the total predicted by the model and the total predicted by the planners. In Fig 12. We have picked out a typical example of a task type where the model is producing slightly more accurate results than the planner, as in task types 1 and 3. From this graph we can see that although the prediction is slightly off the planned, the planners were further off for most of the week. The predicted line does tend to follow the trend of the actual across the whole week also.

In Fig. 13 we can see an example where the predicted accuracy is significantly superior to the planner, as seen in task types 2 and 6. The predicted line follows the actual almost exactly for the first half of the week, although there is some discrepancy on Thursday. In general however the trend does follow that of the actual values, hitting near the exact for most of the week.

In Fig. 14 we show the breakdown for the problem task types, that of 4 and 5. Here we see that the prediction is significantly off on Saturday and doesn't tend to follow the trend for the remainder of the week, merely taking the average point. Interestingly, although we achieve greater accuracy than the planners in task type 4 we can also see this Saturday discrepancy. In this case however the average value through the week is better than what the planners were plotting and thus we achieved a greater accuracy. The difference in these cases would suggest that the problem arises from the fact that unlike the other 4 task types the values achieved on a Saturday are significantly different to the rest of the week. Thus these two may be better served with a Weekday and Weekend model split. However this problem should be removed once enough weeks of data becomes available to move to using a different model for each day of the week.

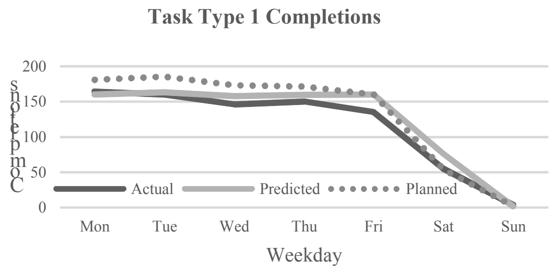

Fig. 12. Typical task type where the model is slightly more accurate

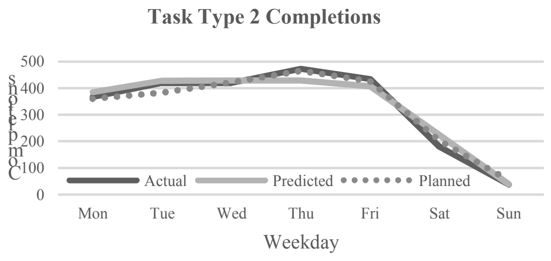

Fig. 13. Typical task type where the model is significantly more accurate

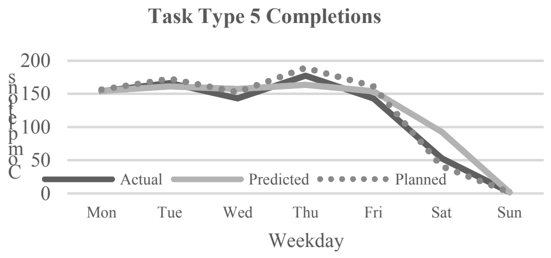

Fig. 14. Problem task type example

VII. Conclusions

In this paper we investigated methods to accurately predict completions in a tactical planning situation. We defined the predictive planning problem and identified some possible methods that could be used to solve it. From these methods we found the NN suitable to achieve our goals. We describe several scenarios we used to develop and refine the model through rigorous experimentation, aiming to achieve the goal of creating a system that is at least as accurate as the current manual planning process but would complete the task in far less time.

The initial general model developed in this paper produced very acceptable results. We managed to produce a very high accuracy whilst also ruling out the need to utilize any techniques, such as correlation filtering, which would add an extra calculation burden to the training process and also complicate the storage and use of these trained models within the final application. We also ruled out the need to use past completion data, which would begin to utilize predicted data points as the model is run on subsequent days.

Moving onto the specific scenario where there are less inputs available to the model, we have managed to further improve the initial model to the point where it is producing a better overall accuracy than the current planning process despite having a limited dataset. Improving upon the current accuracy has met the goal of this paper as it will allow the use of this model to automate this part of the planning process, allowing improved decision making as the planners involvement is raised from that of constructing the plan to exploring different scenarios. A few problem task types were highlighted however, but this is likely to be solved during future development and even with these task types in the problem scenario the overall model accuracy still outperformed that of the current planning techniques.

The models presented here are also suitable to support automated planning optimization, being used to evaluate the effects of planning decisions such as applying overtime to increase capacity, recruiting additional resources, etc. This is a focus for further work.

With the real world problem presented in this paper solved there is also now scope for further work investigating the suitability of alternate techniques, such as support vector machines [19].

ACKNOWLEDGMENT

We would like to thank the reviewers of this paper for their useful and helpful comments that have assisted in fine tuning this paper and have also helped to guide our hand towards possible areas of future work.

REFERENCES

[1] C. Voudouris, “Defining and Understanding Service Chain Management,” in *Service Chain Management*, Springer, 2008, pp. 1-17.

[2] G. Owusu, P. O'Brien, J. McCall and N. F. Doherty, Transforming Field and Service Operations, Springer, 2013.

[3] S. Shakya, S. Kassem, A. Mohamed, H. Hagras and G. Owusu, “Enhancing Field Service Operations via Fuzzy Automation of Tactical Supply Plan,” in *Transforming Field and Service Operations*, Springer, 2013, pp. 101-114.

[4] C. Voudouris, G. Owusu, R. Dorne and D. Lesaint, Service Chain Management, Springer, 2008.

[5] G. Owusu, Anim-Ansah, G and M. Kern, “Strategic Resource Planning,” in *Service Chain Management*, Springer, 2008, pp. 35-49.

[6] R. Ainslie, S. Shakya, J. McCall and G. Owusu, “Optimising Skill Matching in the Service Industry for Large Multi-Skilled Workforces,” in *Research and Development in Intelligent Systems XXXII*, Springer International Publishing, 2015, pp. 231-243.

[7] M. Kern and G. Owusu, “Tactical Resource Planning and Deployment,” in *Service Chain Management*, Springer, 2008, pp. 65-77.

[8] J. Anderson, An Introduction to Neural Networks, MIT Press, 1995.

[9] M. Haga, H. Demuth, M. Beale and O. De Jesús, Neural Network Design (Vol 20), Boston: PWS publishing company, 1996.

[10] N. Kourentzes and S. Crone, *Advances in forecasting with artificial neural networks,* Lancaster University: The Department of Management Science, 2010.

[11] M. Kutner, Applied linear statistical models, vol. 4, Chicago: Irwin, 1996.

[12] M. W. Gardner and S. R. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric environment,* vol. 32, no. 14, pp. 2627-2636, 1998.

[13] G. Zhang, B. Eddy Patuwo and M. Y. Hu, “Forecasting with artificial neural networks: The state of the art,” *International Journal of Forecasting,* vol. 14, no. 1, pp. 35-62, 1998.

[14] J. Heaton, “Encog: Library of Interchangeable Machine Learning Models for Java and C#,” in *Journal of Machine Learning Research*, 2015, pp. 1243-1247.

[15] Y. Chauvin and D. E. Rumelhart, Backpropagation: theory, architectures, and applications, Psychology Press, 1995.

[16] P. Sibi, S. Jones and P. Siddarth, “Analysis of different activation functions using back propagation neural networks,” *Journal of Theoretical and Applied Information Technology ,* vol. 47, no. 3, pp. 1264-1268, 2013.

[17] Ö. KISI and E. Uncuoglu, “Comparison of three back-propagation training algorithms for two case studies,” *Indian journal of engineering & materials sciences,* vol. 12, no. 5, pp. 434-442, 2005.

[18] L. Prechelt, “PROBEN1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules,” Karlsruhe, 1994.

[19] Y. Tian, Y. Shi and X. Liu, “Recent advances on support vector machines research,” in *Technological and Economic Development of Economy*, 2012, pp. 5-33.