IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 13, 2020, accepted May 6, 2020, date of publication May 14, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994519

Simulating the Evolution of Homeless Populations in Canada Using Modified Deep Q-Learning (MDQL) and Modified Neural Fitted Q-Iteration (MNFQ) Algorithms

ANDREW FISHER[1], VIJAY MAGO[1], AND ERIC LATIMER[2]
[1]Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada
[2]Department of Psychiatry, McGill University and Douglas Research Centre, Montréal, QC H4H 1R3, Canada

Corresponding author: Andrew Fisher (afisher3@lakeheadu.ca)

This work was supported in part by Insight grant 21820 from the Social Sciences and Humanities Research Council of Canada.

ABSTRACT It is estimated that over 235,000 Canadians experience homelessness at some point each year. With the emergence of smart cities, it would be beneficial to leverage the processing power of deep learning to assist in the planning and testing of different policies to address this issue. When examining a population of homeless individuals, one can view them as being distributed, at any one point in time, among several possible states: for example, the street or an emergency shelter. Our work aims to provide a means of simulating across these states, including no longer homeless, over time. The probability that an individual will transition from one state to another is called a transition probability. Thus, by creating a matrix of transition probabilities between all of the states, we have a transition probability matrix. If we simply approached this problem by using a mathematical model such as a Markov decision process, we run into the issue of how to accurately adjust the probabilities to produce realistic results. Ideally, we would have a model that can reasonably modify them based on real-life data. To do this, we introduce two modified deep learning algorithms; modified deep q-learning (MDQL) and modified neural fitted q-iteration (MNFQ). These algorithms dynamically produce a set of transition probability matrices for each week of the year. We discuss the modifications we made to these algorithms to adapt to the homelessness problem and create our simulation. After training our model on high resolution, weekly data, we will show that when running it on a low resolution data set that spans 3 years, our model is able to achieve a relative percent difference from the final population of 12.5%. The end result is a model that can be further improved over time with real world data to provide realistic results.

INDEX TERMS Simulation, machine learning, homelessness, policy making, planning.

I. INTRODUCTION

A. THE PROBLEM OF HOMELESSNESS

Homelessness is a source of growing concern across Canada as well as in most developed countries [1], with numbers increasing in most Canadian cities [2]–[7] and internationally [8]–[10]. It is estimated that on any given night, about 567,715 people are homeless in the United States [8], and 35,000 in Canada [11]. Homelessness is associated with worse physical and mental health [12], [13], increased mortality [14], greater criminal behavior and victimization [15], and high health and criminal-justice-related costs [16].

The associate editor coordinating the review of this manuscript and approving it for publication was Miltiadis Lytras.

The causes of the rise in homelessness are not completely understood but certainly include rising income inequality [17] together with rising rents in many areas [18]- leading to a growing shortage of affordable housing. At an individual level, many factors predispose a person towards homelessness, including low educational attainment, joblessness or low income, poverty, mental illness, and substance abuse [19], [20]. In between these macro- and micro-level factors are institutional arrangements such as lack of adequate supports for people who were previously homeless or who are at risk of homelessness who leave the youth protection system, prisons, and hospitals [19]. These factors interact with each other in a complex manner [19].

In recognition of the complexity of the phenomenon, numerous policies and programs, operating at different levels, have been put in place to address homelessness. These have included increasing the availability of affordable housing (whether through building new affordable housing units or providing low-income individuals with rent supplements); helping individuals who have become homeless regain permanent housing, through a variety of more or less intensive and short- or long-term supports, notably Housing First, which offers individuals a combination of a rent subsidy and the long-term support of a mobile clinical team [21], [22]; and an array of primary, secondary and tertiary prevention measures [23]. The complexity of the phenomenon, together with the wide range of possible remedial and preventive measures, means that addressing homelessness effectively is challenging, and experts disagree on many aspects of the policies that should be pursued in a given city or geographical area.

B. COMPUTER SIMULATION MODELING TO HELP ADDRESS HOMELESSNESS

Computer simulation modeling offers a possible decision support tool to address homelessness. This approach has often been used to try to gain deeper insight into complex problems of many kinds [24]Ḷimited attempts in this direction have been made until now with regards to homelessness, however. These attempts can be classified, to date, into 4 groups: (a) economic models calibrated to individual cities [25]; (b) entirely mathematical models that do not incorporate any city- or area-specific empirical data; (c) mathematical models based on a survey of empirical results found in the literature [20]; (d) statistical models that relate the number of homeless individuals in an area to a number of other area-specific variables. Among these, only the first has the potential to simulate the effects of alternative specific policies on the number and composition of homeless individuals in an area. The one study of this type that we have identified is, however, entirely focused on the housing market and the effects of alternative housing subsidy mechanisms; It does not take into account programs to help homeless individuals access housing, such as Housing First; nor does it distinguish among the different states that homeless people can find themselves in, such who experienced a one-time, brief episode in a homeless shelter, or someone who has been alternating for years between sleeping in street locations and in shelters. This was the main objective of our project.

C. RESEARCH CHALLENGES

Constructing such a fine-grained model poses two main challenges: how to structure the model, and where to get the data to calibrate it. We chose to structure it as a Markov model, with a cycle length of one week, in which individuals can be located in any one of 8 states: street location, shelter, transitional housing, substance abuse treatment center, hospital, prison, not homeless (but previously so), or deceased. Individuals transition from one state to the next on the basis of a set of transition probabilities. The capacity of shelters and transitional housing in a city, by age group (25 and under or over 25) and gender, is input into the system, and occupancy of these resources cannot exceed 100%. (The other types of institutions mostly serve non-homeless individuals and their capacities can thus, for our purposes, be considered unconstrained.)

To derive the transition probabilities, we used data from the Montreal site of a large Canadian study, the At Home/Chez Soi trial (refs). This site followed 463 initially homeless individuals over up to two years, reconstructing their day-by-day housing trajectories. Thus it provides data at a resolution sufficient to enable simulation of each individual in a cohort separately. This is necessary given our desire to incorporate shelter and transitional housing capacity constraints. Recruitment for this study took place, however, between 2009 and 2011, and the sample was not designed to be representative of the homeless population as a whole. Furthermore, the service system in Montreal has evolved since the early 2010s, notably with the progressive addition of Housing First programs with a combined capacity of several hundred.

In order to calibrate the model, we had access to data from Montréal's March 24 2015 and April 24 2018 point-in-time homelessness counts, which provide not only the number of homeless individuals at those dates, but detailed data on their locations and demographic characteristics. We wanted, therefore, to base ourselves on the transition probabilities to reproduce the evolution of the number and composition of the homeless population from the first count to the second. A method is needed for adjusting the transition probabilities so that they fit the point-in-time data. As some individuals left the system (due to death or exiting homelessness), others also enter it, becoming homeless for the first time and, in some cases, remaining homeless for the long term. This also needs to be represented in the model. To this end, some data were available from another survey of homeless individuals conducted in Montreal five months after the first point-in-time count. These data provided information, for individuals who were homeless on August 24, 2015, on whether they were homeless on March 24 2015, and if so, in what type of location. Our basic approach was to use two modified q-learning algorithms to adjust the transition probabilities so as to be able to reproduce, as closely as possible, the 2018 count data starting with those from 2015.

D. SUMMARY OF OUR ALGORITHMS AND CONTRIBUTIONS

Our simulation model works with two, modified deep-learning algorithms; the originals being deep q-learning [26] and neural fitted q-iteration [27]. Originally, the algorithm would start in an initial state, perform an action, and observe the new state it transitioned to [26], to determine a reward [28]. Instead, with our simulation, the action performed is simply determining which *new state* to transition to; the algorithm is picking the new state based on the transition probability matrix. The reward for our algorithm is

determined by what the current populations in the simulation are and what the final populations should be (from the training data) after the simulation has ended.

For neural fitted q-iteration, the original algorithm worked with deep q-learning by performing an offline update to determine the best action to perform in the algorithm's current state. It is offline in a sense that all *previous* changes are considered when doing it [27]. The main modification we made here was that, instead of determining the best action, the algorithm calculates an *offset* to modify the new q-values by, based on previous changes. This offset helps lower or increase the q-values, which are later interpreted as the transition probability matrices for each state transition pair. In a sense, the result is the best action that the algorithm determines to ensure realistic transitions between states.

To summarize, we made the following modifications to both the deep q-learning and neural fitted q-iteration algorithms:

- Removed the ''action'' step
- Modified the purpose of the reward term
- Modified the neural fitted q-iteration to make it more versatile for other applications

E. ORGANIZATION OF THE PAPER

In the next section, we will examine related work to our proposed model. We will discuss recent papers on homelessness issues, modelling approaches, and machine learning in social sciences to show how our family of algorithms has been beneficial in other implementations. In the subsequent section, we will look at the methodology of our approach and describe each modified algorithm in detail. Lastly, we will provide a discussion of our results and propose future work to improve our model. This section will also include results when our model was applied to Montreal's 2015 and 2018 homelessness count data, using transition probabilities derived from the Montreal site of the At Home/Chez Soi project [29], [30].

II. RELATED WORK

A. PREVIOUS EFFORTS TO MODEL THE PHENOMENON OF HOMELESSNESS

As mentioned earlier, previous efforts to model the phenomenon of homelessness can be classified into 4 groups: (a) economic models calibrated to individual cities [25]; (b) entirely mathematical models that do not incorporate any city- or area-specific empirical data; (c) mathematical models based on a survey of empirical results found in the literature; (d) statistical models that relate the number of homeless individuals in an area to a number of other area-specific variables.

Authors of [25] provide, to our knowledge, the only example of the first group of efforts. We describe it in more detail as it is the only one that in some ways approximates what we are trying to do. The authors used a general equilibrium model of the housing market to examine policies to reduce homelessness, calibrated for four California cities. Theirs is a model of the housing market, in which ''dwelling units filter through a quality hierarchy. . . and in which households of various income levels choose among these discrete types.'' Households may choose to opt out of the housing market and thus become homeless. They are more likely to do so if available housing is unaffordable to them. Thus, increases in homelessness are driven by changes in rents. They concluded that ''a very large fraction of homelessness can be eliminated through increased reliance upon well-known housing subsidy policies'', in particular, rent subsidies [25].

Authors of [31], [32] provide examples of the second type of model. As these models do not incorporate any empirical data, they are of limited value as a decision-support tool for a particular city. Alone to our knowledge in the third group, [20] applied a well-established technique called fuzzy cognitive maps to analyze the impact of social factors on homelessness. Their macro-level model, which was calibrated using information extracted from the literature, was able to reasonably represent reality for a range of scenarios. The direction and strengths of the relationships between concepts included in the map approximated their action in reality. Education emerged as having the greatest force in the model. Again, however, the model remains general, and not useful as a decision-support tool for a particular city [20].

Finally, the fourth group is comprised, again, of a single effort. [33] led the development of the ''Homelessness analytics initiative'' (http://homelessnessanalytics.org/). This web site compiles a large amount of data on homeless counts in the many US areas where these are now regularly carried out, and social (e.g., crime) and health (e.g., county-level life expectancy) indicators as well as other contextual factors (e.g., fair market rents). The web site also provides access to a set of forecasting models. These models are based on regression analyses of homeless counts against social indicators and other predictor variables. However, once again, these statistical models, being based on commonly available social and economic indicators, are of limited usefulness in estimating the effects and costs of alternative policies in a particular city or area.

B. MATHEMATICAL MODELLING

A mathematical modeling presented by Zhang, T., Xie, S., and Rose, O. at the Winter Simulation Conference in 2018 where the main issue they addressed was that previous attempts at automating batching in job shops were falling short of their goals in real-life environments. The batching, here, is done when several jobs are processed *simultaneously* [34]. The approach was due to the fact that the typical job was too complex in nature and not fit for a stochastic environment; that is, the jobs lacked a linear pattern that could be easily modelled. Practitioners also preferred real-time batching where the scheduling was considered as more of a decision-making problem. To solve this problem, the paper introduces a sequential decision-making process using Markov decision processes.

Another mathematical model by Batata, O., Augusto, V., and Xie, X. focused on optimizing care resources by

predicting the burnout in a caregiver to admit them to respite services before hospitalization is needed [35]. The main issue was that previous attempts fell short when it came to actually predict the number of patients needing to be admitted. Respite care is a new service that aims to help decrease the burnout risk in caregivers. When this is not taken care of in time, the caretaker eventually needs hospitalization, which is costly. This paper talks about the need to be able to predict burnout in a caregiver and admit them to respite services before hospitalization is needed.

Authors of [35] attempted to use the addition of machine learning as well as Markov chain transition matrices to create a dynamic burnout model with two states for the caregivers: emergency and normal. By having the ability to see what a caregiver's next state will be, the system could efficiently decide whether or not they should be admitted into respite services. From their experiments, Batata et al's model performed best using a *neural network*; the objective was to minimize the number of hospitalized individuals. One major shortcoming of this work, however, was that the Markov chain was strictly built using only burnout data without considering attributes specific to the caregiver and their patient.

C. MACHINE LEARNING AND SIMULATION

Authors of [28] introduced a new approach using machine learning and simulation to optimize the dose calculation for radiotherapy. This approach involved two steps: an agent-based simulation of vascular tumor growth and a q-learning algorithm to optimize the radiation dosages. The optimal outcome when combining these two steps is to achieve a cure for tumor with minimal side effects. The researchers noted that many studies had been done in the area of radiotherapy simulation, and optimization but not in optimizing radiotherapy based on simulation. Due to a lack of real data, the agent-based simulations noted before were ran to help the generation of synthetic data needed for the optimization. The outputs allow interfering in the simulations to examine different scenarios. Since this area has not been researched much, it is hard to note any shortcomings. Two points of consideration: there was no real-life experiment, and this approach uses a process they called inverse planning. This process is aided by a computer to test (simulate) different treatment plans before any physical experimentation is done [28].

An important theme amongst recent papers in this area involved q-learning. More precisely, neural fitted q iteration (NFQ) [27]. The main issue that this algorithm addressed was an issue that arose with multi-layer perceptrons when training them. During the training process, when modifying a parameter change in one area of the network, the algorithm has the potential to influence other values later on in the network. The approach inadvertently destroys the effort done so far in other regions and leads to long learning times or, worst case, not learning at all. In order to solve this limitation, when updating the q-value functions, the algorithm offers previous knowledge explicitly as well by storing all previous experiences in terms of state-action transitions in memory.

To implement their approach, the algorithm uses what's called an off-line update rule. In other words, the algorithm considers an entire set of transition experiences- contrary to traditional q-learning, which uses an on-line update rule. This strategy gives the advantage of applying advanced supervised learning methods to the network. For example, [27] uses Rprop, which is very efficient and very insensitive to the learning parameters. NFQ falls under the fitted q iteration family of algorithms; it is a memory-based method used to train q-value functions based on multi-layer perceptrons. By exploiting generalization, the algorithm is able to achieve a high level of data efficient learning. The paper provides three real-life scenarios that are quite diverse in specifications. This shows that the algorithm is applicable to a wide variety of tasks and can work well with real-life scenarios.

III. METHODOLOGY

A. DATA SOURCE

We are currently working with data from the At Home/Chez Soi project [29], [30] to train our algorithm. Researchers gathered the data of individuals who entered and exited different states (street, shelter, etc.) at different points in time over a year. Interviewers tracked these movements using retrospective questionnaires administered every 3 months. This method of data collection makes it possible to generate transition matrices on a week-to-week basis. Since each person will be simulated individually, we can accommodate the exact information of data available to us. The individuals can each be considered a data point for the algorithm.

B. DATA PRE PROCESSING

Transition probabilities were estimated from data collected at the Montreal site of the At Home/Chez Soi project [36]. This was a large (n = 2,148) randomized controlled trial of Housing First for homeless individuals with mental illness, compared to usual services, conducted in five Canadian cities: Vancouver, Winnipeg, Toronto, Montreal, and Moncton. Housing First offers individuals experiencing homelessness immediate access to a choice of subsidized, rental market apartments, together with the support of a mobile team of mental health and other professionals [30]. Study participants were all people with mental illness, who had been homeless for varying lengths of time and who expressed an interest in being housed. They were recruited between October 9, 2009 and May 31, 2011. They were interviewed at 3-month intervals for up to 2 years. A questionnaire was used to reconstruct places they had been each night since the previous interview [37].

C. MACHINE LEARNING BASED ALGORITHMS

At a basic level, we can model the simulation using a Markov decision process with a transitional probability matrix. This matrix is calculated by looking at the number of individuals who go from one state to another, then calculating the probability that this will occur based on the total population in that

initial state. An example set of transition probabilities can be seen in Figure 2. Using such a model would essentially be a simplified, mathematical representation of reinforcement learning [34]- a common approach when training machine learning models.

By using a mathematical model, we would need to monitor the outputs and see if any changes are needed for the model. This process can become cumbersome when working with large datasets, such as a population of people, as the values would, initially, need to be updated constantly to produce accurate results. To address this problem, we will create a machine learning model that, although requiring a small amount of reinforcement learning, will mostly be unsupervised when training on real-life data. The small amount of reinforcement learning here is from the value of an *offset* and if it is minimizing from previous iterations. An example visualization of these probabilities and how they may change from one time period to the next can be seen in Figure 1, which shows the values of a basic set of transition probabilities from one time period to the next. The arrows in this diagram represent the transition from one state to another, with their probability noted as the p value. We demonstrate our model on data from the At Home/Chez Soi project [29], [30] to show how it learns and modifies the transition probability matrices over time.

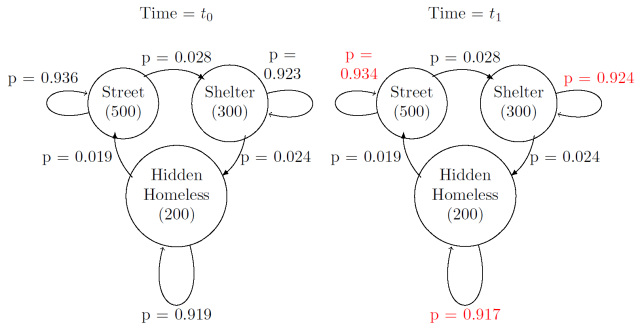


FIGURE 1. A simplified example set of transition probabilities between states and how they may change in a machine learning model. Note that this is not showing all possible transitions. [38].

Furthermore, we shouldn't assume that a single transitional probability matrix will work when simulating on a higher resolution scale such as week-to-week. To address this assumption, our model will, instead, generate a probability matrix for each week of the year. Although the goal of our model when training is to accurately predict the known final populations, we are also wanting to use it to predict realistic results for future populations where the end population is unknown.

1) HOMELESSNESS SIMULATION

Over time, homeless individuals can transition between many different states of homelessness. For our simulation, these states include street, shelter, hidden homeless, not homeless (but previously were), transitional housing, hospital,

	Street	Shelter	Hidden Homeless	Not Homeless	Transitional Housing	Death
Street	0.936	0.028	0.030	0.006	0.005	0.000
Shelter	0.019	0.923	0.024	0.017	0.017	0.000
Hidden Homeless	0.019	0.029	0.919	0.022	0.011	0.000
Not Homeless	0.006	0.009	0.009	0.969	0.004	0.003
Transitional Housing	0.003	0.008	0.006	0.009	0.974	0.000

FIGURE 2. An example of a simplified set of transition probabilities [38].

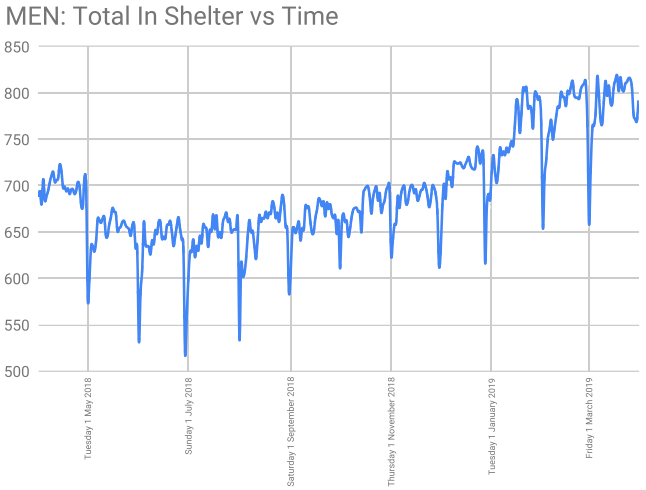


FIGURE 3. The change in Montréal men's shelters over time.

rehabilitation (drug/alcohol), and prison.[1] The *probability* that an individual in an initial state s_i will transition to a new state s_n can be defined as follows in Equation 1:

$$P(s_n|s_i) = \frac{N(s_i, s_n)}{N(s_i)} \tag{1}$$

where $N(s_i, s_n)$ is the number of people transitioning from s_i to s_n and $N(s_i)$ is the number of individuals in s_i. An individual can transition from their initial state to their initial state (no change) or from one state to another state (by the end of week), for instance, street to shelter. Therefore, for the states defined before, we would have 72 probabilities (a matrix of 8 initial states and 9 new states) for each state-to-state transition–excluding death to another state. However, from just analyzing the data alone, we would only take into account the *change in populations*. Other outside factors can also affect this probability. Consider the following graphs where we analyzed the cumulative population of six males (Figure 3) and seven female (Figure 4) shelter populations in Montréal,[2] respectively, over the course of a year as well as the temperature (Figure 5)[3] over that time period. Here, we can clearly see an impact on the shelter population of temperature. This is an example of how simply calculating

[1] Adapted from the At Home/Chez Soi project [29], [30]
[2] Data provided by the city of Montréal
[3] Data retrieved from https://montreal.weatherstats.ca/metrics/ temperature.html

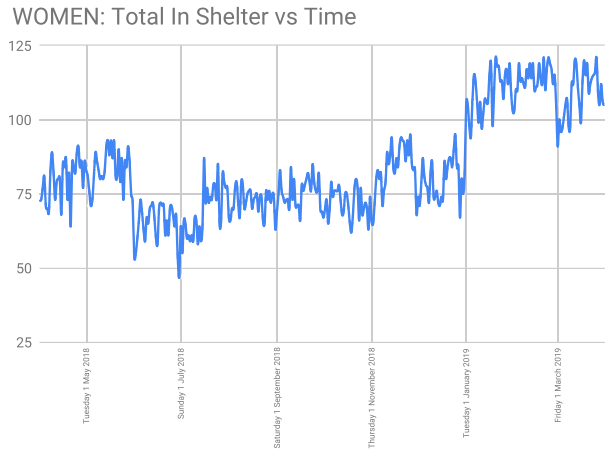


FIGURE 4. The change in Montréal women's shelters over time.

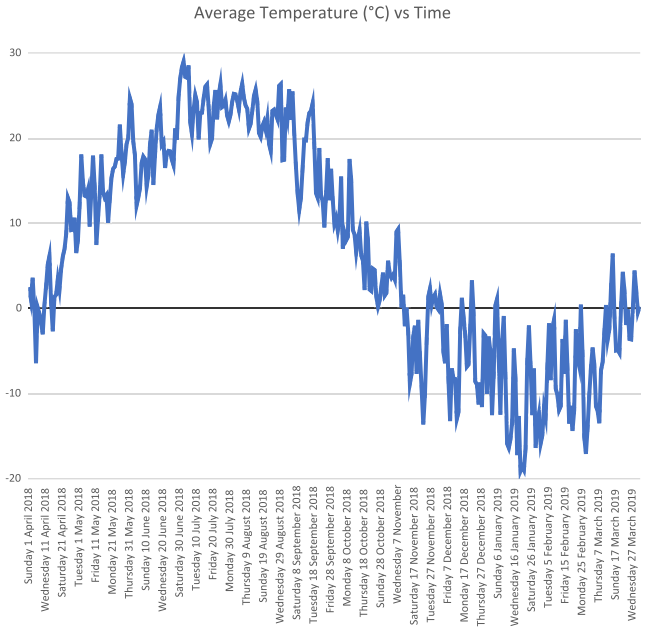


FIGURE 5. Montréal's temperature over time from mid 2018 to mid 2019.

the transition probability matrices for two points in time will not suffice to create an accurate simulation. We propose a model where the transition matrix probabilities are *dynamically updated* based on the current state of the system and what the end result should be when training.

2) NEURAL NETWORKS

To introduce our deep learning algorithms properly, we will first provide a brief introduction to the concept of neural networks. This framework for machine learning, more commonly referred to as artificial neural networks (ANNs), is an attempt to mimic our own cerebral cortex closely [39]. Figure 6 is visual representation of a basic neural network.

Each circle in Figure 6 represents a *neuron* in the network; while each line represents a *connection* between the neurons. The blue neurons represent the *input layer*; the red neurons

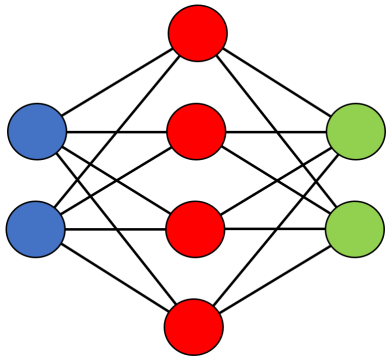


FIGURE 6. A basic neural network.

represent the *hidden layer*, and the green neurons represent the *output layer*. There can be more than one hidden layer, and the output layer does not necessarily need to have the same number of neurons as the input layer; Figure 6 is just a basic representation. Each connection has a parameter known as the *weight* of the connection, and each hidden neuron, as well as output neuron, has a parameter known as its *bias* [39]. When values are presented to the input layer, each neuron has its own input value that is passed along each of its connections to the hidden layer. This value is affected by the weight of the connection it is travelling down. The output of the hidden neuron is affected by all of these values travelling to it, as well as its own bias. These outputs, in Figure 6 for example, would then be passed to the output layer, where the same process would occur to get the final output values. In order to train the network to produce accurate outputs, the difference between the output of the network and what the output should be are first calculated. Then, the network will typically adjust weights by propagating error terms back–a well-known back propagation method [39].

3) MDQ-LEARNING

The base algorithm we are modifying for our purposes is called *q-learning*. It is a form of reinforcement learning where the model (or agent) is trying to determine an optimal action to perform while in a certain state [40]. The way it learns is by receiving a *reward*, positive or negative, for performing said action based on how *optimal* it is for an overarching goal. Consider a simple example known as the *cart-pole* problem [27]. The environment is initialized with a cart that can move left or right and an upright pole attached to it that is slightly offset such that it will fall over if the cart is not moved. The goal of the agent is to keep the pole as upright as possible. If it falls past a certain threshold (such as 15 degrees from the center for example) the agent fails, and the environment is re-initialized. The rewards given in this problem are based on how well the agent corrects the falling of the pole, that is, how close the pole stays to 0 degrees from the center [27].

In deep q-learning, the algorithm aims to learn an optimal rule or policy for a Markov decision process. It does not require a model to process the data; it simply takes

a starting state, performs an action, and observes the new state reached. The learning here also comes from a reward given for reaching the new state, which the algorithm aims to maximize. When implementing this approach in a neural network, it could be defined as the *distance* from the desired output that it wants to reach.

This algorithm uses a *target network* and a *prediction network* to determine the *loss* of the system where the output of the prediction is trying to converge with the target. Every c number of epochs, the architecture of the *target network* is updated with that of the *prediction network* being trained during this process. In order to maximize the reward, the algorithm modifies its q-values, which are used when deciding which action to perform as noted before. Given a sufficient amount of time, the algorithm can get close to converging on the desired output [26], [28], [40].

We will be interpreting the q-values as transition probabilities for each state-to-state transition. One of our algorithm's main difference from deep q-learning is that it only uses *one neural network*; loss will be calculated by looking at the previous output to see how it is converging. Another difference is how we will process the state transitions. Unlike deep q-learning, our data does not necessarily have an *action* that leads to a new state; it is a direct transition.

In order to simulate each member of the population accurately, we will introduce a *sub-epoch* that occurs on every *epoch* in the algorithm. This sub-epoch will run each member, through the network, with their state and action, producing an ideal q-value for them *individually*. To get the reward for this new q-value, consider the following in Equation 2:

$$R_s(s_i, s_n) = 1 - \frac{Q_e(s_i, s_n)}{Q(s_i, s_n) + Q_e(s_i, s_n)} \tag{2}$$

where $Q_e(s_i, s_n)$ is the calculated q-value at the *current* epoch and $Q(s_i, s_n)$ is the q-value for that state-action pair as of the *previous* epoch.

In deep q-learning, the goal is to maximize this reward. Since our goal is, instead, to produce the best *q-value* for this transition, we can see an obvious problem with this formula. If the network needs to lower the value of a probability, it will be difficult with this reward formula as it only produces positive values. This topic will be addressed in the MNFQ-Learning section.

We will introduce a simple approach in determining the new states in a population commonly known as a *roulette wheel*. Consider the following in Equation 3:

$$Q(s_i, s_n) > \sum_{j=1}^{2} Q(s_i, s_{n_j}) > \sum_{j=1}^{x} Q(s_i, s_{n_j}) \tag{3}$$

where x is the total number of new states. This definition assumes that the q-values have been normalized between 0 and 1. To select the new state, we can simply generate a random number between 0 and 1 then see where it falls in the range (denoted by the > symbols).

For example, if you had three probabilities 0.25, 0.35, and 0.4, the wheel would look like the following in Equation 4:

$$Wheel = 0.25 > 0.60 > 1.00 \tag{4}$$

Then, the model would generate a random number, for example 0.55. Next, the model would identify if it falls between 0 and 0.25. If it does not, it would go to the next range which is 0.25 and 0.60; which would be a match. So, the selected action would be action 2.

Once all of the sub-epochs have ran, we can calculate the new q-values for the algorithm. Consider the following in Equation 5:

$$Q'(s_i, s_n) = Q(s_i, s_n) + [1 - \frac{\sum_{j=1}^{N} R_j(s_i, s_n)}{N}]\eta \tag{5}$$

where $Q'(s_i, s_n)$ is the new q-value, $Q(s_i, s_n)$ is the previous q-value; N is the size of the population; R_j is the reward for each member of the population in that transitioned from state s_i to state s_n, and η is the learning rate. This equation essentially takes the average of the rewards from the maximum reward, multiplied by the learning rate, as the value to update the q-value by. After all of the updates have occurred, we will then normalize the results to ensure they still fall in between 0 and 1.

The last consideration is how the network is trained. Consider the following in Equation 6 for the error:

$$E_{out} = \frac{\sum_{j=1}^{x}[Q'_j(s_i, s_n) - \frac{\sum_{k=1}^{N} Q_{e_k}(s_i, s_n)}{N}]}{x} \tag{6}$$

where x is the number of q-values in the table and Q_{e_k} is the outputted q-value for each member of the population that transitioned from state s_i to state s_n. We are calculating the average difference between each new q-value and the average q-value that the population outputted. This error will then be applied to the output neuron of the network so that back-propagation can be performed.

In deep q-learning, the goal is for the prediction network and target network to converge. Since our implementation only uses one network, we will simply look at how the q-values are converging from one epoch to the next. Consider the following definition in Equation 7:

$$Loss = \frac{\sum_{j=1}^{x}[\frac{\sum_{k=1}^{y}[Q'(s_{i_j}, a_{n_k}) - Q(s_{i_j}, a_{n_k})]^2}{y}]}{x} \tag{7}$$

Here, we are getting the average of the average of the differences between the new q-values and the old q-values squared where x is the number of states and y is the number of actions.

To input into the states to the network, we simply create an array of them and assign their index in the array as their value to the network. Therefore, the network takes two inputs: the index of the state of the person and the index of the state to which they are transitioning. The output is an optimal q-value based on the current state of the model for the current member of the population. Our proposed model has a network with

four hidden layers having 4, 8, 16, and 24 nodes respectively. We do give the option of modifying the size and count of the hidden layers here, but this is our suggestion from testing the model. For example, if we had the states street, shelter, and hospital, street would be recognized as state 0, shelter as state 1, and hospital as state 2 to this neural network. As for an initial transition matrix, we suggest manually analyzing the input data first and calculating a matrix to speed up the algorithm's training. These probabilities can be calculated using Equation 1 in the homelessness simulation section.

4) MNFQ-LEARNING

In neural fitted q-learning (NFQ), the main concept is to update the q-values in deep q-learning using an off-line approach where all previous experiences are considered [27]. These experiences are defined as the original state, action taken, and the resulting state. The reasoning for this approach is because when a q-value is updated it could, inadvertently, affect somewhere else in the table and require an update there as well.

Since the model is using an off-line approach (all previous experiences are considered instead of just the current experience), it will determine the best course of action for the current q-value being updated in an attempt to avoid this. This approach results in fast training times with a minimal amount of input [27], [34]. For the homelessness problem, however, using the modified deep q-learning algorithm (MDQL) we defined, the action is simply the new state. Therefore, the purpose of this algorithm will, instead, be to *assist* MDQL in updating the q-values on each epoch to a reasonable value.

In order to determine an accurate q-value (or transition probability for our purposes) for the next epoch, this algorithm will calculate an offset to add onto the calculated q-value from MDQL. Consider the following in Equation 8:

$$Q'(s_i, s_n) = Q(s_i, s_n) + [1 - \frac{\sum_{j=1}^{N} R_j(s_i, s_n)}{N}]\eta + \delta Q(s_i, s_n)\eta \quad (8)$$

This is the same equation as the new q-value in the MDQL algorithm but with the offset, $\delta Q(s, a)$, that MNFQL will calculate. The probability (or q-value) is accurate based on whether or not it is *realistic* for the training data.

The way this offset will be calculated for each $Q(s_i, s_n)$ is by considering all of the previous q-values outputted by the MDQL algorithm for that state-action pair, the current population in the new state, the known end population for new state, and the weeks left in the simulation. To increase accuracy with the q-values, we will use a separate neural network for each state-new state pair. Each network is individually trained with the data for its respective state-new state pair, before training the primary MDQL algorithm as a result. For example, if this problem was applied to the probability set in Figure 2, the result would be $5 \times 6 = 30$ networks.

Each network will take the following inputs: each q-value previously calculated for that pair individually, the population in the new state, weeks left in the simulation, and the current q-value. The output of the network will be the end population for the new state. The q-value that gives the lowest error will be defined as α, and the output error from that will be used for back propagation in the network. This error, E, is simply the difference between what the network outputs and the known end population. For the offset, consider the following in Equation 9:

$$\delta Q(s, a) = \frac{\alpha}{\sum_{j=1}^{x} Q(s_i, s_n)}(\frac{-E}{N_{s_n}}) \quad (9)$$

where N_{s_n} is the known end population size for the new state. Since the network is taking the total population in the new state, we need to times by the best q-value divided by the sum of all states with this new state (the summation would exclude the state-new state pair that α is a part of and add α on instead) to get an estimated percentage that this state transition will contribute to the final population. This equation will then give us the relative, percent difference in the end population, multiplied by this.

For the rewards in MDQL, we propose the following modification to the formula as shown in Equation 10:

$$R_s(s_i, s_n) = \sigma Q(s_i, s_n) \quad (10)$$

where $\sigma Q(s_i, s_n)$ is the *immediate* value from $\delta Q(s, a)$. This uses the same formula for $\delta Q(s, a)$, but α (previously the q-value that gives the lowest error) is, instead, the q-value calculated for the current individual. If the error from this is less than 20%, back propagation is performed on the state, new state network.

IV. AN EXAMPLE

Consider a set of ten (10) individuals that can only be in one of three of the following *homelessness* states: shelter, street, and hospital. These individuals are divided up as shown in Figure 7.

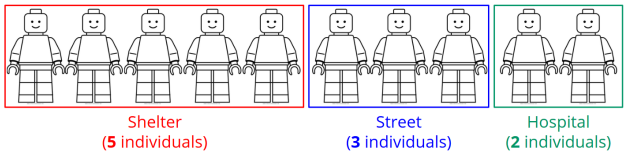


FIGURE 7. An example homeless population.

For each of the states, the algorithm "sees" the following: shelter as 0, street as 1, and hospital as 2. Therefore, for the population in figure 7, the algorithm would "see" it as an array as follows: {0, 0, 0, 0, 0, 1, 1, 1, 2, 2}.

A. INITIALIZATION

Consider the population in the previous section and, for the sake of simplicity, say that a year has 4 weeks (or quarters). This example will have two (2) years of data as shown in Figure 8.

The algorithm would "see" the data in Figure 8 as follows:

Year 1

	Shelter	Street	Hospital
Week 1	5	3	2
Week 2	4	4	2
Week 3	4	2	4
Week 4	3	5	2

Year 2

	Shelter	Street	Hospital
Week 1	6	3	1
Week 2	5	4	1
Week 3	5	1	4
Week 4	2	8	0

FIGURE 8. An example dataset.

YearOne = {0, 0, 0, 0, 0, 1, 1, 1, 2, 2}, {0, 0, 0, 0, 1, 1, 1, 1, 2, 2}, {0, 0, 0, 0, 1, 1, 2, 2, 2, 2}, {0, 0, 0, 1, 1, 1, 1, 1, 2, 2}

YearTwo = {0, 0, 0, 0, 0, 0, 1, 1, 1, 2}, {0, 0, 0, 0, 0, 1, 1, 1, 1, 2}, {0, 0, 0, 0, 0, 1, 2, 2, 2, 2}, {0, 0, 1, 1, 1, 1, 1, 1, 1, 2}

Since the algorithm is presented with three (3) states (shelter, street, and hospital) and four (4) quarters in a year, it will initialize the transitional probabilities for each quarter (4 sets in total) as shown in Figure 9 (these were randomly generated with the highest probability given to stay in the same state).

New State

Current State	Shelter	Street	Hospital
Shelter	0.72	0.22	0.06
Street	0.32	0.58	0.10
Hospital	0.27	0.22	0.51

= 9 probabilities = P

FIGURE 9. An example transition probability matrix.

The next step of the initialization is to create one (1) neural network for MDQ-Learning (used to simulate) and P (number of probabilities) individual neural networks for MNFQ-Learning (used to train the model).

B. TRAINING: INITIAL LOOP

The algorithm starts by cycling through the data for each week of each year. Let's follow the first week of year one:

Input population = {0, 0, 0, 0, 0, 1, 1, 1, 2, 2}

This population is passed to the MDQ-Learning algorithm to simulate each person *individually*. We will follow the first person who is in the shelter state; consider an example transitional probability matrix for this week of the year (week 1) in Figure 10.

New State

Current State	Shelter	Street	Hospital
Shelter	0.72	0.22	0.06
Street	0.32	0.58	0.10
Hospital	0.27	0.22	0.51

FIGURE 10. Week 1's Transition Probability Matrix.

Since the individual we are following is in the shelter state, we are only concerned with the probabilities that they will transition to other states *from* their initial state. Therefore, transitioning to the shelter has a 72% probability of occurring, street a 22% probability, and hospital a 6% probability. With that in mind, the new state for this individual is determined using a weighted roulette wheel based on these probabilities as shown in Figure 11.

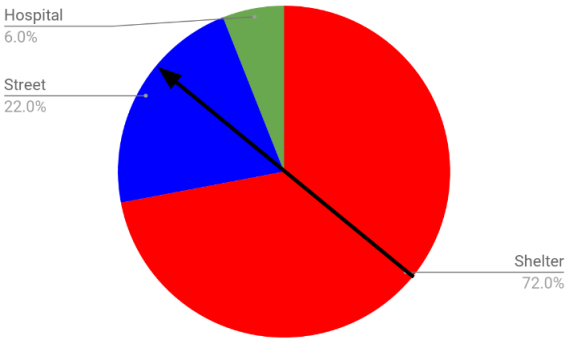


FIGURE 11. A sample, weighted roulette wheel.

The wheel is "spun" (a random number is generated) to determine which state to transition this individual to. The new state is recorded (for the individual we are following, street), and input into the MDQ neural network with the previous state (for the individual we are following, shelter) to get a *recommended* transition probability for the individual. This output will be defined as $Q_e(s_i, s_n)$, where s_i is the initial state, s_n is the new state, and Q_e is the output. Consider Figure 12 which shows a sample MDQ network.

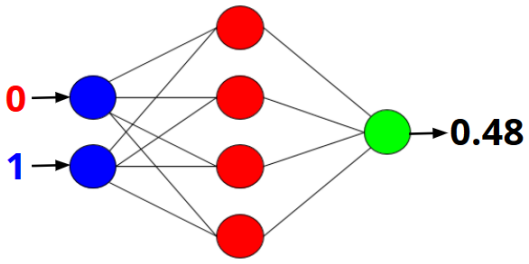


FIGURE 12. An example MDQ network.

Next, with the MNFQ neural network for the current individual's transition (in this case, the shelter → street network), it takes the following inputs:

(1) The transition probability calculated for the current individual, normalized with the current value (from Figure 12, 0.48)
(2) The current population in this new state (in this case, after the transition, there's now 4)
(3) The weeks/epochs left to the next population (in this case, the next set of available data is 1 quarter away)

The goal is to see the output's error from the end population (start of next week) in this new state (for street, 4). This error will be referred to as E.

The error is used to determine an immediate *reward* for choosing this transition for this individual with the previously described formula:

$$\delta Q(s, a) = \frac{\alpha}{\sum_{i=1}^{x} Q(s_i, s_n)} \left(\frac{-E}{N_{s_n}}\right) \tag{11}$$

where α is the calculated transition probability (from the previous output, 0.48), N_{s_n} is the end population (for street, 4), and the summation is of all the transition probabilities for each state → street probability. Since the goal of the network is to output the total population in the new state, we calculate the current individual's calculated transition probability divided by the sum of all transitional probabilities to this new state (state → street) to get an estimated percentage that the current state transition (shelter → street) will contribute to the overall, relative percent error of the final population (in this example, street population). The goal of the algorithm is to *minimize* this value.

This "reward" as well as the calculated transition probability is recorded for each individual. If the error from before, E, is less than 20%, the MNFQ neural network (in this case, the shelter → street network) is trained by performing back-propagation. Once all individuals have been processed, the algorithm uses each MNFQ network to determine an *offset* to add to each transition probability in order to better fit the data.

Consider the same equation described for the "reward" value previously but this time, however, α is instead the calculated transition probability, from each individual, that gives the *lowest error* for this transition. The rest of the equation variables are identical. This value is then used in the previously described equation:

$$Q'(s_i, s_n) = Q(s_i, s_n) + [1 - \frac{\sum_{j=1}^{N} R_j(s_i, s_n)}{N}]\eta \tag{12}$$

where $Q'(s_i, s_n)$ is the new transition probability, $Q(s_i, s_n)$ is the previous transition probability, N is the size of the population in the new state s_n, R_j is the "reward" for each member of the population in that transitioned from state s_i to state s_n, and η is the learning rate. Once all transition probabilities for the current week have been updated accordingly, each row is normalized such that it adds up to one (1). An example of this can be seen in Figure 13.

Before

Current State \ New State	Shelter	Street	Hospital	
Shelter	0.85	0.43	0.10	= 1.38
Street	0.22	0.72	0.15	= 1.09
Hospital	0.16	0.32	0.76	= 1.24

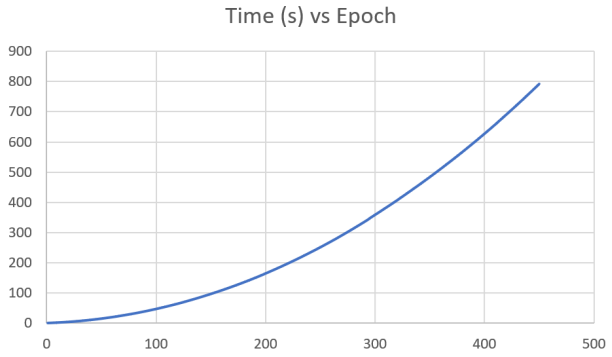
After

Current State \ New State	Shelter	Street	Hospital
Shelter	0.62	0.31	0.07
Street	0.20	0.66	0.14
Hospital	0.13	0.26	0.61

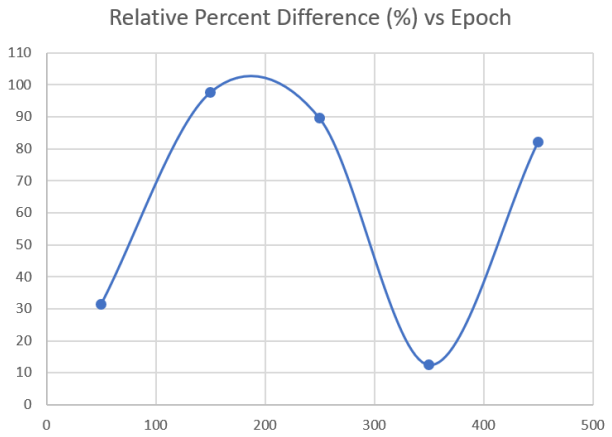
FIGURE 13. A sample transition probability normalization after a training epoch has finished.

V. PERFORMANCE EVALUATION

As discussed in our research challenges, we are faced with a lack of data. Using the At Home/Chez Soi project [29], [30] we created a data set that showed the state for each individual in Montréal at 117 different time-points (or weeks). It should not be assumed that every individual had a state recorded for every time-point; however, this issue did not introduce any complications as our model trains on a week-to-week basis. We first trained our model with this data to produce transition probability matrices for each week of the year. This process required us experimenting with the number of epochs and learning rate to determine the best combination for an optimal output. The three learning rates we tested with were 0.1, 0.01, and 0.001. After numerous rounds of testing, the lowest relative percent differences from the final population we achieved were 12.9%, 12.5%, and 66.5% respectively.



(a) A graph showing the overall processing time for training our model after 450 epochs



(b) A graph showing the relative percent difference of our model after 50, 150, 250, 350, and 450 epochs

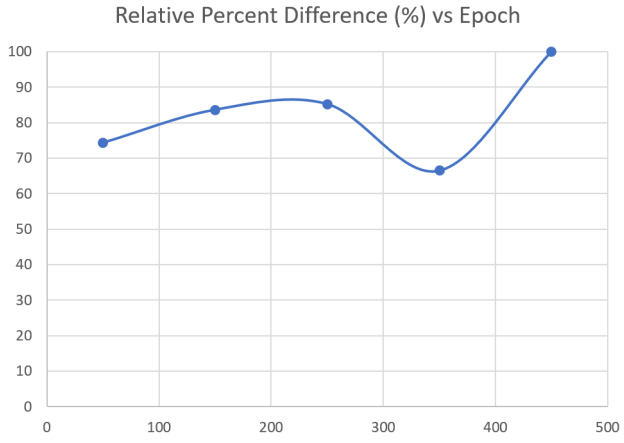
FIGURE 14. Figures for a learning rate of 0.01.

Consider Figure 14 that shows the total processing time for our maximum run of 450 epochs at a learning rate of 0.01. It should be noted that for the other two learning rates, the processing time was very similar to this graph. We can see that the time increases exponentially as the epochs are increased. It should be noted that the model is currently using the CPU as the hardware accelerator.[4] Once the model was trained,

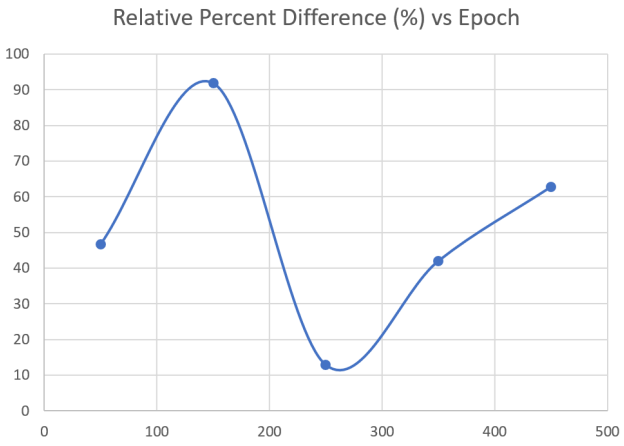
[4]The time results in this graph were from an Intel Core i7-7700K processor

it produced 52 unique transition probability matrices for each week of the year.

Next, we tested the model with 2015 to 2018 homeless state counts from the city of Montréal. Since our model transitions the population on each epoch, we consider one epoch to be equal to one week. Therefore, we ran the populations in 2015 through the trained model for 156 epochs. Consider Figures 14 and 15 that show the relative percent difference of the final output for this learning rate, after 50, 150, 250, 350, and 450 epochs.



(a) Learning rate of 0.001



(b) Learning rate of 0.1

FIGURE 15. Figures for our relative percent difference with learning rates 0.001 and 0.1.

It shows that the relative percent difference of our model followed an almost sinusoidal form. This observation is interesting as lower epochs may be suffice to produce an accurate output. In this instance, at 50 epochs, our model had an relative percent difference of 31.39%. Consider figure 16, which shows the best results for each learning rate's best run. At a learning rate of 0.1, we can see that the not homeless and transition housing states were very closely predicted. For the other states, however, we do have some differences that need to be improved. We will discuss implementing a Markov model in the future works section to compensate for this.

State	Simulated Results	Actual Results
Not Homeless	43620	102979
Transitional Housing	3433	1231

(a) Best results with a learning rate of 0.001

State	Simulated Results	Actual Results
Not Homeless	106636	102979
Transitional Housing	810	1231

(b) Best results with a learning rate of 0.01

State	Simulated Results	Actual Results
Not Homeless	100335	102979
Transitional Housing	1391	1231

(c) Best results with a learning rate of 0.1

FIGURE 16. Figures for our results with each learning rate.

	S1	S2	S3
S1	0.25	0.75	0.00
S2	0.00	0.25	0.75
S3	0.00	0.00	1.00

FIGURE 17. A simple transition probability matrix.

To evaluate our model further, we created an experiment to generate *synthetic data* based on an input transition probability set. This step was implemented because of lack of real data available to us at this current time. Consider the simple transition probability matrix for three states in Figure 17.

Our experiment assumes that this matrix is valid for all 52 weeks in the year. This probability matrix forces the population to eventually transition to state S3 and be unable to transition out of it. The weekly data generated from our experiment follows the matrix perfectly. For example, if 100 individuals are in S1, the next week will place 25 in S1 and 75 in S2. To lengthen the time that the resulting simulation will take to converge, our experiment initially places more people in the state that is less likely to be reached. For the transition probability table in Figure 17, with a total population of 100, this resulted in S1 starting with 59 individuals, S2 with 33, and S3 with 8. From Figure 18, we can see that this dataset (containing 52 weeks in total) converges quite quickly. Furthermore, consider Figure 19 that shows the state populations over each week.

With our model, a perfect result would be a graph that looks like Figure 19 after having been trained with the synthetic dataset then asked to simulate a starting population equal to week 0 in Figure 18 for one year (52 weeks). In Figures 20 and 21, our model was run for 75 epochs with randomly initialized and provided transition probabilities respectively. These figures also include the loss over epochs

WEEK	S1	S2	S3
0	59	33	8
1	14	53	33
2	3	24	73
3	0	9	91
4	0	2	98
5	0	0	100
6	0	0	100

FIGURE 18. A simple dataset generated from Figure 17 probabilities.

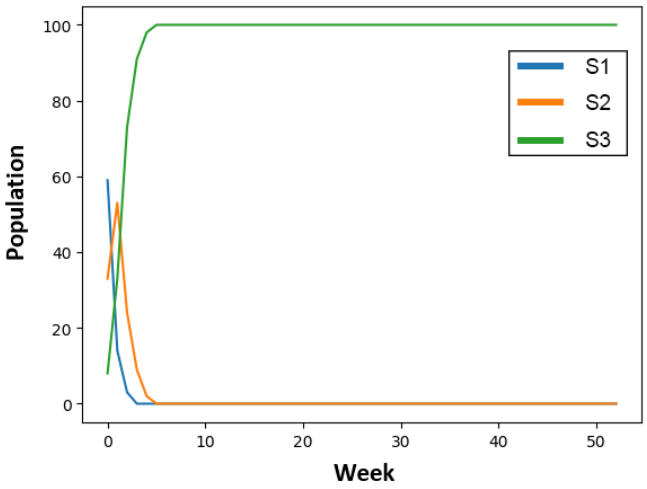
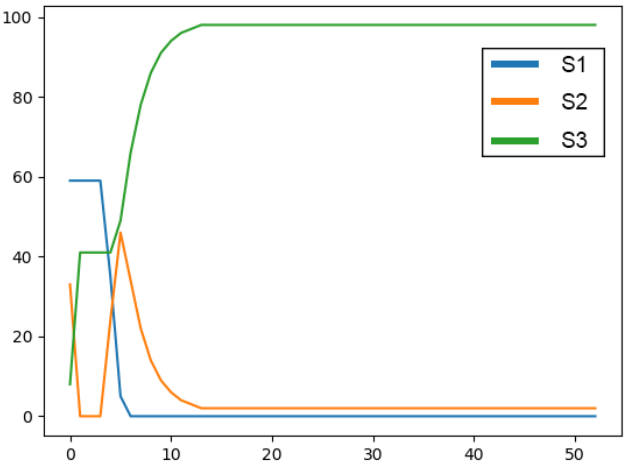


FIGURE 19. A graph of the state populations over weeks from the dataset shown in Figure 18.



(a) A graph produced as per Figure 19

(b) The loss over epochs of this model

FIGURE 20. A model using a learning rate of 0.01 and trained for only 75 epochs.

as well to show how the probability set converged as the model was trained (i.e.. a lower loss means a smaller difference between transition matrices from the previous epoch).

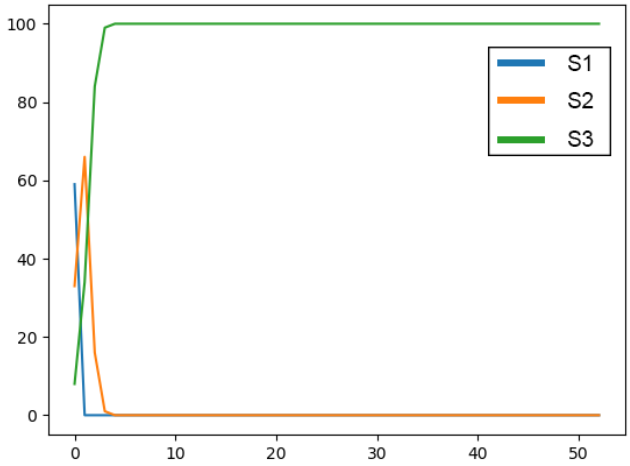
To compare the results with our previous testing, the model was ran again but with the transition probability provided to it; consider Figure 21. The resulting graph is very close to Figure 19 but has an interesting curve in the loss over epochs, which could be attributed to the fact that the model knew the exact transition probabilities (the loss dropped significantly at the start) and started to overfit the model as seen with the lack of curves and sharp lines in Figure 19.

VI. DISCUSSION AND FUTURE WORKS

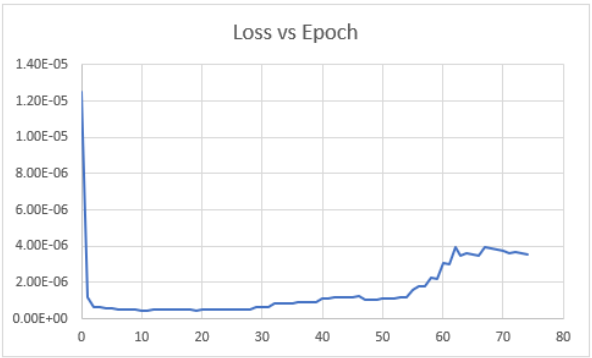
As we have shown, our model is able to learn very quickly with a relatively small amount of data to produce accurate results. Although the test data was very low resolution, we are confident in our simulation as it learned from real data and was able to produce a simulation that ended with a low error to our desired results. One of the aspects we can improve here is our overall processing time. Although using CPU as the hardware accelerator was shown to be considerably quick, processing with a GPU would only speed up the algorithm even more.

One may argue that reinforcement learning may be too complex for this problem. This method is applicable where there's the need to learn simultaneously (1) the dynamics of the system, and (2) a control policy suitable for achieving some externally-imposed goals. Because the purpose of this research is to provide policy makers and planners with a means of predicting the future populations of each homelessness state, we feel that it is necessarily complex. It can be argued that the algorithm could indeed be used to achieve externally imposed goals based on how it is used. For example, a planner may add or remove shelters based on the output, which the algorithm will then adapt to in order to provide realistic outputs based on the real-word data on which it was trained.

Therefore, we disagree that using a simpler, modelling strategy (such as a MLP classification network using multi-class cross-entropy loss) would be better for this problem. Although the initial training could be considered as such, the end result is a model that will predict the population distributions over time. This could, again, be argued as classifying individuals into different population groups as a function of

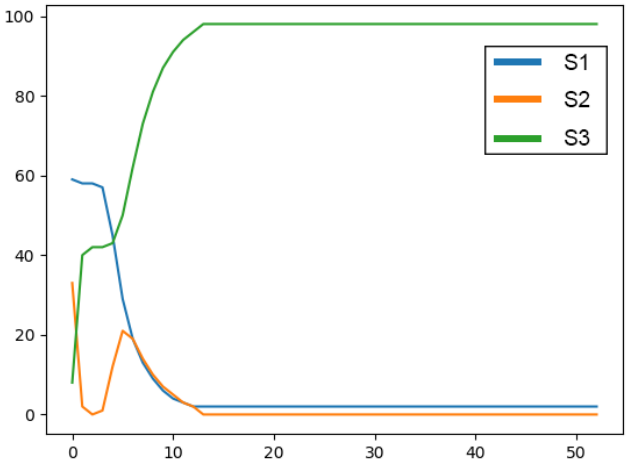


(a) A graph produced as per Figure 19

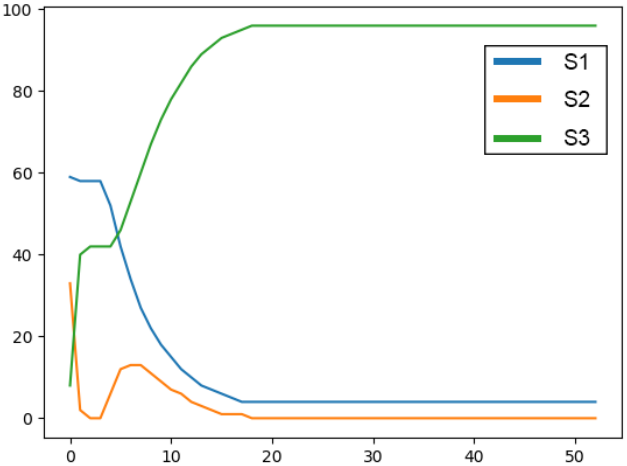


(b) The loss over epochs of this model

FIGURE 21. The same model as Figure 20 but with the transition probabilities provided to it.



(a) The model without onehot encoding



(b) The model with onehot encoding

FIGURE 22. A model using a learning rate of 0.1 and trained for 100 epochs.

time but a problem arises when determining a proper function of the current state and any desired auxiliary information. By using our deep learning methods as proposed in the paper, the algorithm is, instead, learning this function on its own and removing that complexity which would be difficult to accurately create otherwise.

Another aspect to improve on with our model is the process of transitioning individuals to new states. For the shelter state, for example, an individual wouldn't be able to transition to it if the city they are in has no availability in their shelters. To implement this feature we would look at adding a Markov decision process to replace the roulette wheel approach. By using real data from shelters in the city the user is targeting, we would get realistic populations in this example state.

As noted previously, our MDQ network takes the index of an individual's current state, as well as the individual's new state. An interesting approach would be to convert these indices to *onehot encoding* instead as they may be considered as being ranked or ordered otherwise. Consider the example shown previously where Shelter was 0, Street was 1, and Hospital was 2. For this example, these would be converted to 100, 010, and 001 respectively. From our testing with the synthetic dataset described previously, the differences were quite subtle. However, to see how it would affect extreme cases, consider a model that has a high learning rate (0.1) and trains for 100 epochs. As shown in Figure 22, the differences are not substantial but one can see that without onehot encoding, the result is indeed closer to the exact result shown in Figure 19.

VII. CONCLUSION

By creating a model to simulate a population of homeless individuals accurately, we can provide policy makers and planners with a means of predicting the future populations of each homelessness state. If we simply used a mathematical model, it would be a difficult task to create as we would constantly need to revise the model manually to produce realistic results for each transition probability.

Instead, we propose a model that leverages the processing power of deep learning to achieve this result. With this model, we can input known homelessness data and have the probabilities revised dynamically to produce more accurate results. Our two algorithms, modified deep q-learning and modified neural fitted q-learning, work together to achieve

the same effect and input the resulting probabilities into a Markov decision process to transition the population between states.

The main challenge of our research was the lack of high resolution homelessness data, which is important as our model needs to train on and produce realistic results. One way we accommodated this consideration was by simulating each member of the population *individually*. From our performance analysis, we were able to see our approach produce an accurate model with a relative percent difference of 12.5% on a low resolution data set that was entirely different from our training data set. Furthermore, with a synthetic dataset, we applied our algorithm to a higher quality source and confirmed that it indeed produces accurate results.

ACKNOWLEDGMENT

The authors would like to thank Dr. A. Rao for assisting in the proof-reading of our article.

REFERENCES

[1] T. Lancet, "A shared future for all: Let's talk about homelessness," *Lancet*, vol. 391, no. 10117, p. 179, Jan. 2018.
[2] *Street Needs Assessment Results 2013*, City Toronto, Toronto, ON, Canada, 2013.
[3] *Street Needs Assessment Results 2018*, City Toronto, Toronto, ON, Canada, 2018.
[4] *Vancouver Homeless Count 2015*, M. Thomson Consulting, Vancouver, BC, Canada, 2015.
[5] *Vancouver Homeless Count 2018*, Urban Matters CCC BC Non-Profit Housing Assoc., Vancouver, BC, Canada, 2018.
[6] E. Latimer, J. McGregor, C. Méthot, and A. Smith, "Dénombrement des personnes en situation d'itinérance à Montréal le 24 Mars 2015/i count Montreal: Count and survey of Montréal's homeless population on march 24, 2015," Ville de Montréal, Montréal, QC, Canada, Tech. Rep., 2015.
[7] *Dénombrement des Personnes en Situation D'itinérance sur L'île de Montréal le 24 Avril*, Ville de Montréal et CIUSSS du Centre-Sud-de-l'Île-de-Montréal, Montréal, QC, Canada, 2019.
[8] *Estimated Number of Homeless People in the United States From 2007 to 2019 Statista*, Statista, Hamburg, Germany, 2020.
[9] *Homelessness in Australia Up 14% in Five Years, ABS Says; in the Guardian*, Guardian Media Group, London, U.K., 2018.
[10] *Europe and Homelessness: Alarming Trends*, Second Overview Housing Exclusion Eur., Brussels, Belgium, P. F-TFA, 2017, vol. 1.
[11] S. Gaetz, E. Dej, and T. Richter, "The state of homelessness in Canada 2016," Can. Homelessness Res. Netw., Toronto, ON, Canada, Tech. Rep., 2016.
[12] S. Fazel, V. Khosla, H. Doll, and J. Geddes, "The prevalence of mental disorders among the homeless in western countries: Systematic review and meta-regression analysis," *PLoS Med.*, vol. 5, no. 12, p. e225, 2008.
[13] S. W. Hwang, "Homelessness and health," *CMAJ*, vol. 164, no. 2, pp. 229–233, 2001.
[14] J. S. Roncarati, T. P. Baggett, J. J. O'Connell, S. W. Hwang, E. F. Cook, N. Krieger, and G. Sorensen, "Mortality among unsheltered homeless adults in Boston, Massachusetts, 2000–2009," *JAMA Internal Med.*, vol. 178, no. 9, pp. 1242–1248, 2018.
[15] L. Roy, A. G. Crocker, T. L. Nicholls, E. A. Latimer, and A. R. Ayllon, "Criminal behavior and victimization among homeless individuals with severe mental illness: A systematic review," *Psychiatric Services*, vol. 65, no. 6, pp. 739–750, Jun. 2014.
[16] E. A. Latimer, D. Rabouin, Z. Cao, A. Ly, G. Powell, T. Aubry, J. Distasio, S. W. Hwang, J. M. Somers, V. Stergiopoulos, S. Veldhuizen, E. E. M. Moodie, A. Lesage, and P. N. Goering, "Costs of services for homeless people with mental illness in 5 Canadian cities: A large prospective follow-up study," *CMAJ Open*, vol. 5, no. 3, pp. E576–E585, Jul. 2017.
[17] L. Farha, *We Can't Talk About Inequality Without Talking About Homelessness*. London, U.K.: The Guardian, 2016.
[18] C. Glynn and A. Casey, "Homelessness rises faster where rent exceeds a third of income," Zillow, Seattle, WA, USA, Tech. Rep., 2018.
[19] M. Piat, L. Polvere, M. Kirst, J. Voronka, D. Zabkiewicz, M.-C. Plante, C. Isaak, D. Nolin, G. Nelson, and P. Goering, "Pathways into homelessness: Understanding how both individual and structural factors contribute to and sustain homelessness in Canada," *Urban Stud.*, vol. 52, no. 13, pp. 2366–2382, Oct. 2015.
[20] V. K. Mago, H. K. Morden, C. Fritz, T. Wu, S. Namazi, P. Geranmayeh, R. Chattopadhyay, and V. Dabbaghian, "Analyzing the impact of social factors on homelessness: A fuzzy cognitive map approach," *BMC Med. Informat. Decis. Making*, vol. 13, no. 1, p. 94, Dec. 2013.
[21] K. Jones, P. W. Colson, M. C. Holter, S. Lin, E. Valencia, E. Susser, and R. J. Wyatt, "Cost-effectiveness of critical time intervention to reduce homelessness among persons with mental illness," *Psychiatric Services*, vol. 54, no. 6, pp. 884–890, Jun. 2003.
[22] A. Levitt, "Review of housing first: The pathways model to end homelessness for people with mental illness and addiction," *Psychiatric Rehabil. J.*, vol. 35, no. 2, pp. 156–157, 2010.
[23] S. Gaetz and E. Dej, *A New Direction: A Framework for Homelessness Prevention*. Toronto, ON, Canada: Canadian Observatory on Homelessness Press, 2017.
[24] J. Sterman, "Business dynamics: Systems thinking and modeling for a complex world," Massachusetts Inst. Technol. Eng. Syst. Division, Cambridge, MA, USA, Tech. Rep., 2000.
[25] E. T. Mansur, J. M. Quigley, S. Raphael, and E. Smolensky, "Examining policies to reduce homelessness using a general equilibrium model of the housing market," *J. Urban Econ.*, vol. 52, no. 2, pp. 316–340, Sep. 2002.
[26] T. Eden, A. Knittel, and R. Van Uffelen, *Reinforcement Learning*. Accessed: May 15, 2019. [Online]. Available: https://www.cse.unsw.edu.au/~cs9417ml/RL1/algorithms.html
[27] M. Riedmiller, "Neural fitted Q iteration—First experiences with a data efficient neural reinforcement learning method," in *Proc. Eur. Conf. Mach. Learn.*, 2015, pp. 317–328.
[28] A. Jalalimanesh, H. S. Haghighi, A. Ahmadi, and M. Soltani, "Simulation-based optimization of radiotherapy: Agent-based modeling and reinforcement learning," *Math. Comput. Simul.*, vol. 133, pp. 235–248, Mar. 2017.
[29] T. Aubry, P. Goering, S. Veldhuizen, C. Adair, J. Bourque, and J. Distasio "A randomized controlled trial in five Canadian cities of the effectiveness of housing first with assertive community treatment for persons with serious mental illness and a history of homelessness," *Psychiatric Services*, vol. 67, no. 3, pp. 81–275, 2016.
[30] V. Stergiopoulos, S. W. Hwang, A. Gozdzik, R. Nisenbaum, E. Latimer, D. Rabouin, C. E. Adair, J. Bourque, J. Connelly, J. Frankish, L. Y. Katz, K. Mason, V. Misir, K. O'Brien, J. Sareen, C. G. Schütz, A. Singer, D. L. Streiner, H.-M. Vasiliadis, and P. N. Goering, "Effect of scattered-site housing using rent supplements and intensive case management on housing stability among homeless adults with mental illness: A randomized trial," *J. Amer. Med. Assoc.*, vol. 313, no. 9, p. 905, Mar. 2015.
[31] B. O'Flaherty, "Individual homelessness: Entries, exits, and policy," *J. Housing Econ.*, vol. 21, no. 2, pp. 77–100, 2012.
[32] P. J. Fowler, P. S. Hovmand, K. E. Marcal, and S. Das, "Solving homelessness from a complex systems perspective: Insights for prevention responses," *Annu. Rev. Public Health*, vol. 40, no. 1, pp. 465–486, Apr. 2019.
[33] *Access Critical National, State, and Local Information About Homelessness and Related Factors*, Homelessness Anal. Initiative, 2020.
[34] T. Zhang, S. Xie, and O. Rose, "Real-time batching in job shops based on simulation and reinforcement learning," in *Proc. Winter Simulation Conf. (WSC)*, Gothenburg, Sweden, Dec. 2018, pp. 3331–3339.
[35] O. Batata, V. Augusto, and X. Xie, "Mixed machine learning and agent-based simulation for respite care evaluation," in *Proc. Winter Simulation Conf. (WSC)*, Dec. 2018, pp. 2668–2679.
[36] E. Latimer, D. Rabouin, C. Méthot, C. McAll, A. Ly, and H. Dorvil, *At Home/Chez Soi Project: Montréal Site Final Report*. Calgary, AB, Canada: Mental Health Commission Canada, 2014.
[37] P. N. Goering, D. L. Streiner, C. Adair, T. Aubry, J. Barker, J. Distasio, S. W. Hwang, J. Komaroff, E. Latimer, J. Somers, and D. M. Zabkiewicz, "The at Home/Chez soi trial protocol: A pragmatic, multi-site, randomised controlled trial of a housing first intervention for homeless individuals with mental illness in five Canadian cities," *Brit. Med. J. Open*, vol. 1, no. 2, Nov. 2011, Art. no. e000323.
[38] A. Fisher, V. Mago, and E. Latimer. *Homelessness Simulation Using Modified Deep Q-Learning Algorithms*. [Online]. Available: https://www.researchgate.net/publication/341215636_Homelessness_Simulation_Using_Modified_Deep_Q-Learning_Algorithms

[39] J. Burger. *A Basic Introduction To Neural Networks*. Accessed: May 15, 2019. [Online]. Available: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html
[40] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992. Accessed: May 15, 2019.

ANDREW FISHER enrolled at Brandon University in 2014 as a Dual-Credit Student to pursue the B.Sc. degree with a major in computer science and a minor in mathematics and physics. Throughout his degree, he worked as a Marking Assistant in the Computer Science Department and assisted in the laboratory periods for first year classes. Towards the start of his third year, he was recommended by one of his professors to a company called Heartland Software Solutions, who a Brandon University graduate, Dr. R. Bryce, had been running for almost 15 years. He was offered a position as a Remote Server Administrator and a Software Developer which he quickly accepted. He has been exposed to many programming languages as a result, including PHP, C#, C++, Java, R, and more. Throughout the remainder of his degree, he continued marking and maintained high GPAs as he was placed on the dean's honor list from the second year on. During his final year at university, he worked with a Professor named Dr. Srivastava, as well as Dr. Bryce, as a part of his honors degree. They worked on modifying the MQTT protocol's standard to include geolocation data. Dr. Bryce had implemented some of this work in a C# code base and Andrew's work was to expand on this as well as implement it in a Java code base to eventually create an Android app. After working on the project throughout the year, the group submitted an article detailing work to the 2019 IEEE 89th Vehicular Technology Conference in Kuala Lumpur, as well as an article detailing the Android app Andrew created to the 2019 42nd International Conference on Telecommunications and Signal Processing in Budapest, Hungary. Both articles were accepted with appreciation from peer reviewers. Today, he has been enjoying his time at Lakehead University under the supervision of Dr. V. Mago. He is working on his thesis that involves creating a homelessness simulation using machine learning. He has been working on developing two algorithms for this so far and enjoys the work very much. He looks forward to graduating in December 2020 with a specialization in artificial intelligence and pursuing the Ph.D. degree in January 2021.

VIJAY MAGO received the Ph.D. degree in computer science from Panjab University, India, in 2010. In 2011, he joined the Modeling of Complex Social Systems Program at the IRMACS Centre of Simon Fraser University. He is currently an Associate Professor with the Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada, where he teaches and conducts research in areas, including big data analytics, machine learning, natural language processing, artificial intelligence, medical decision making, and Bayesian intelligence. He has served on the program committees for many international conferences and workshops. In 2017, he joined Technical Investment Strategy Advisory Committee Meeting for Compute Ontario. He has published extensively (more than 50 peer reviewed articles) on new methodologies based on soft computing and artificial intelligence techniques to tackle complex systemic problems, such as homelessness, obesity, and crime. He serves as an Associate Editor for IEEE Access and *BMC Medical Informatics and Decision Making* and as a Co-Editor for the *Journal of Intelligent Systems*.

ERIC LATIMER received the Ph.D. degree. He is currently a Research Scientist with the Douglas Research Centre and Professor, Department of Psychiatry, McGill University, Montréal, Canada. A health economist by training, he has carried out research on mental health services and services for homeless people for more than 25 years. He has carried out multiple studies on evidence-based practices for people with severe mental illness and homeless people, including Assertive Community Treatment, supported employment, Housing First and the strengths model for case management. He was lead researcher for the Montreal site of the CAN$ 110 million At Home/Chez soi pan-Canadian trial of Housing First, and the lead economist for the national study. He has also led the analysis of surveys of homeless individuals in the province of Quebec, Canada. With more than 100 peer-reviewed publications, his work has been recognized by awards including the 2015 Psychosocial Research Canada's "Pioneer award" for research on recovery for people with severe mental illness, and the 2017 McLaughlin-Gallie Visiting Professorship, Mental Health in Disadvantaged Populations, from the Royal College of Physicians and Surgeons of Canada.