Machine learning to inform tunnelling operations: recent advances and future trends

Brian Sheil, Stephen Suryasentana, Michael Mooney and Hehua Zhu

Brian B. Sheil
Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK.
Email: brian.sheil@eng.ox.ac.uk
ORCID: 0000-0002-1462-1401

Stephen K. Suryasentana
Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK.
Email: stephen.suryasentana@eng.ox.ac.uk

Michael A. Mooney
Department of Civil and Environmental Engineering, Colorado School of Mines, 1500 Illinois St.,
Golden, CO 80401, USA.
Email: mooney@mines.edu

Hehua Zhu
Department of Geotechnical Engineering, College of Civil Engineering Tongji University,
Shanghai 200092, China.
Email: zhuhehua@tongji.edu.cn

Initial submission, 27 April 2020

Main text wordcount: 4980

Figures: 11

Tables: 3
ABSTRACT

The proliferation of data collected by modern tunnel boring machines (TBMs) presents a substantial opportunity for the application of machine learning (ML) to support the decision-making process on site with timely and meaningful information. The observational method is now well-established in geotechnical engineering and has a proven potential to save time and money relative to conventional design. ML advances the traditional observational method by employing data analysis and pattern recognition techniques, predicated on the assumption of the presence of enough data to describe the modelled system’s physics. This paper presents a comprehensive review of recent advances and applications of ML to inform tunnelling construction operations with a view to increasing their potential for uptake by industry practitioners. This review has identified four main applications of machine learning to inform tunnelling, namely TBM performance prediction, tunnelling-induced settlement prediction, geological forecasting and cutterhead design optimisation. The paper concludes by summarising research trends and suggesting directions for future research for ML in the tunnelling space.
INTRODUCTION

Rapid urbanisation points to the use of underground space as one of the most viable, sustainable and efficient means of delivering new services and transport in congested urban areas. The use of trenchless technology in infrastructure construction is growing in popularity for its cost and environmental savings compared to conventional open excavation techniques. In these obstructed underground spaces, optimising the performance of tunnelling operations is critical to ensure safe and economical construction while also preventing damage to existing infrastructure both above and below ground.

Traditionally, tunnelling contractors have used ‘rules of thumb’ and empiricism in addition to more formal design calculations. While simplified design calculations play an important role in tunnel design and construction, optimising tunnelling operations is technically challenging due to their dependence on several complex factors such as site geology, tunnel boring machine (TBM) operational parameters and tunnel geometry (O’Dwyer et al. 2018, 2019, Phillips et al. 2019). Although a significant body of research conducted over the last thirty years has greatly enhanced our understanding of these effects and their influence on tunnelling operations, the literature contains many examples where static ‘rule-based’ design methods fail to provide satisfactory prediction of field behaviour e.g. Barla et al. (2006), Choo and Ong (2015), Sheil et al. (2016).

The proliferation of data collected by modern TBMs presents a substantial opportunity for the application of machine learning (ML) to support the decision-making process on site with timely and meaningful information (Sheil et al. 2020). The observational method is now well-established in geotechnical engineering and has a proven potential to save time and money relative to conventional design (e.g. Royston et al. 2020). ML advances the traditional observational method by employing data analysis and pattern recognition techniques, predicated on the assumption of the presence of enough data to describe the modelled system’s physics. While Shreyas and Dey (2019) present a high level overview of machine learning approaches, the focus of this paper is on the potential application of ML to support decision-making in tunnelling operations.
techniques for tunnelling settlement and performance prediction, a more comprehensive
review of recent advances and applications of ML to inform tunnelling construction
operations is necessary to increase their potential for uptake by industry practitioners. To
this end, this review has identified four main applications of machine learning to inform
tunnelling, namely TBM performance prediction, tunnelling-induced settlement prediction,
geological forecasting and cutterhead design optimisation. The paper concludes by
summarising research trends and suggesting directions for future research for ML in the
tunnelling space.

MACHINE LEARNING MODELS

Overview

The practice of ML has experienced immense recent growth, driven by advances in
computational performance, sensing technology and data storage. Artificial intelligence (AI),
ML and deep learning are three terms often used interchangeably to describe software that
behaves in an intelligent manner. ML is a subset of AI which provides systems with the
ability to automatically learn and perform certain tasks without being explicitly programmed.
Deep learning is a further subset of ML which uses a specific ML algorithm called 'deep'
artificial neural networks, with many hidden layers, to learn from large amounts of data.

A drawback of many supervised learning techniques is the requirement for a large database
of high-quality information to accurately capture the physics of the modelled system. The
size of the dataset required for the training process is highly dependent on the type of ML
technique adopted, its intended role (e.g. interpolation, optimisation, forecasting) and the
complexity of the input-output relationship being modelled. This section provides a brief
overview of ML techniques commonly applied to tunnelling operations.
An artificial neural network (ANN) is an information processing paradigm that draws inspiration from the operation of the human brain. A network consists of multiple interconnected layers of neurons, comprising a layer of input neurons, one or more layers of ‘hidden’ neurons that perform operations on the data, and a layer of output neurons. Transformation of the input data is performed by the artificial neurons through the application of a nonlinear function (known as the activation function) of the sum of weighted inputs (see Fig. 1). In its simplest form (a feedforward neural network), data travels in one direction – from input to output. After each complete iteration, termed ‘epochs’, the network output values are compared to the target values to produce an error measurement. Feedback of the error through the network, known as ‘backpropagation’, is a nonlinear optimisation process which adjusts the weight and bias of each connection towards reducing the value of the cost function. In this paper the ‘architecture’ describes the network structure in the form \( n_f \ldots n_{hi} \ldots n_o \) where \( n_f, n_{hi} \) and \( n_o \) are the number of features, neurons in hidden layer \( i \) and outputs respectively.

An alternative network form is a recurrent neural network (RNN) wherein connections between units form a directed cycle. This allows the network to maintain information in ‘memory’ over time and therefore use historical calculations to determine outputs. Long short-term memory (LSTM) is a type of RNN that uses a ‘memory cell’ that can store information for long periods of time. A set of ‘gates’ are used to decide whether information is stored in the memory cell, when information from the memory cell is deployed in the network or when information is removed from the cell altogether (i.e. forgotten).

Fuzzy logic

Fuzzy logic (FL) involves the integration of expert knowledge and experience into a fuzzy inference system using fuzzy ‘If-Then’ rules to model the qualitative aspects of human knowledge. This allows an extension of binary, classic logic to qualitative, subjective and...
approximate situations. Takagi and Sugeno (1985) presented the first systematic investigation of fuzzy modelling. The purpose of a fuzzy inference system is to map inputs to outputs through the application of fuzzy reasoning. Fuzziness is first applied to the inputs to produce a fuzzy set using a 'membership function', \( \zeta(x) \), such as the Gaussian membership function:

\[
\zeta = e^{-\frac{(x-c)^2}{2\sigma^2}}
\]

where \( x \) is the input value and \( \sigma \) and \( c \) are the standard deviation and mean of the Gaussian function respectively. The resulting fuzzy set is processed using a set of If-Then rules. The results are subsequently defuzzified to produce 'crisp' outputs.

Adaptive neuro-fuzzy inference systems

Adaptive neuro-fuzzy inference systems (ANFIS) denotes the fusion of neural networks with fuzzy logic principles. The key difference to traditional neural networks is that part or all nodes in the network are modified to be 'adaptive'. This means that the outputs of the network are now dependent on the nodal parameters and the learning rule updates the parameters to minimise a prescribed error measurement. Relationships between variables are defined using fuzzy If-Then rules. ANFIS networks are typically organised in five layers as follows: (a) layer 1 is the input layer comprising the adaptive nodes and node functions and activates the fuzziness of the inputs, (b) layer 2 determines the firing strength of each rule, (c) layer 3 normalises the firing strengths, (d) layer 4 defines the consequence parameters and (e) layer 5 computes the ANFIS outputs by summing the outputs of layer 4.

Fuzzy c-means clustering

Conventional clustering techniques assign data to a cluster without consideration of the extent of its 'belonging' to that cluster. First introduced by Dunn (1974), fuzzy c-means
clustering (FCM) is a clustering approach that allows a datapoint to belong to multiple clusters with varying degrees of membership. This method uses an iterative clustering technique to produce an optimal ‘d’ through the minimisation of an objective function $J_{FCM}$:

$$J_{FCM} = \sum_{i=1}^{n_p} \sum_{j=1}^{n_c} M_{ij}^a \|x_i - d_j\|^2$$  \hspace{1cm} (2)

where $n_p$ and $n_c$ are the number of datapoints and clusters respectively, $M_{ij}$ is the membership matrix, $a > 1$ is a parameter controlling the fuzziness of the system and $\|x_i - d_j\|^2$ is the squared Euclidean distance between observation $x_i$ and cluster centre $d_j$.

Classification and regression trees and random forest

Classification and regression trees (CART) are a non-parametric method that build classification or regression models in the form of a tree structure. At each tree node, a specified number of features are randomly selected and tested to achieve an optimal split of the data. Although decision trees can be highly effective, they are prone to overfitting and are sensitive to the specific dataset upon which they are trained. A robust solution to overfitting is the concept of random forests, first proposed by Breiman (2001). Random forest (RF) is an ensemble learning method that operates by building multiple decision trees and aggregating the results (see Fig. 2). Multiple different training sets (termed bootstrap samples) are generated by sampling with replacement randomly from the original data. This method builds several instances of a decision tree which produces an output $\hat{y}_i$ corresponding to each tree. All individual outputs are then averaged to obtain the final prediction, $\hat{y}$.

Gaussian process regression

A Gaussian process is a collection of random variables of which any finite number follow a joint Gaussian distribution (Williams and Rasmussen, 1996). Gaussian process regression
(GPR) provides a method to perform Bayesian inference about functions in a non-parametric way. One of the key aspects of GPRs is the use of covariance functions which encodes prior assumptions about the functions one wishes to learn (in this case the measured data). This avoids reliance on algebraic mapping between inputs and outputs. The overall aim of the process is to learn a regression model of the form $y = f(x) + \epsilon$, where $f(x)$ is a latent function representing the underlying structure of the data and $\epsilon \sim N(0, \sigma^2)$ is a Gaussian noise term where $\sigma^2$ is the variance of the noise (the symbol `$\sim$' means 'distributed according to'). A GP can be completely described by a mean vector, $\mu(x)$, and covariance function $k(x,x')$ of input pairs $x$ and $x'$ to describe an underlying real process $f(x)$ as follows:

$$f(x) \sim GP(\mu(x), k(x,x'))$$

(3)

where

$$\mu(x) = \mathbb{E}[f(x)]$$

(4)

$$k(x,x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))^T]$$

(5)

Support Vector Machine/Regression

The term 'support vector regression' (SVR) denotes the application of support vector machines (SVM) to regression problems. The $\epsilon$-insensitive approach first proposed by Vapnik (1995) is one of the most widely adopted SVM/SVR approaches in the literature. SVR uses either linear or non-linear kernels to map the input space into a high-dimensional feature space. The most common kernel adopted for this purpose is the radial basis function (RBF):

$$K(x_i, x) = \exp(-\gamma ||x_i - x||^2)$$

(6)

where $\gamma$ is a kernel coefficient. A hyperplane is subsequently constructed in the feature space where the quality of fit to the data is computed using an $\epsilon$-insensitive cost function ($L_\epsilon(y)$) defined as follows (see Fig. 3):
where $x$ is the input data with target values $y$, $f(x)$ is the regression function, and $\varepsilon$ is a user-defined positive value representing the maximum distance between $f(x)$ and $y$ for which there is no loss in the cost function. According to equation (7), only predictions that have residuals greater than $\varepsilon$ are penalised, while predictions with smaller residuals have no effect on the regression equation. Considering a linear function as an example, $f(x)$ can be defined as follows:

$$f(x) = (w \cdot x) + b$$

where $w$ is an adjustable weight vector and $b$ is the bias. The objective is to obtain a function that has the smallest $\varepsilon$ deviation from the target values in the training data and is also as ‘flat’ as possible (by minimising the Euclidean norm $||w||^2$).

**Extreme learning machine**

Extreme learning machine (ELM) is a three-layer neural network i.e. it comprises a single hidden layer (Huang et al. 2004). The novelty of ELM centres around its use of randomly generated hyperparameters for the hidden layer which are not updated during training, unlike conventional neural networks (Huang et al. 2006). This significantly reduces the computational time associated with the learning process and increases the network’s ability to generalise within the trained parameter space. The ELM training process involves the generation and selection of random numbers for the weight and bias matrices for the hidden layer (Huang et al. 2011). Since the number of neurons in the hidden layer is typically much less than the number of training observations, the network is an over-determined linear system. A consequence of this is that the output weight matrix is the only parameter that needs to be optimised during training, which can be undertaken using an ordinary least squares approach.
Particle swarm optimisation

Particle swarm optimisation (PSO) is an optimisation algorithm developed by Kennedy and Eberhart (1995). This approach attempts to mimic interactions in groups of social beings and the sharing of information between the group members (termed ‘particles’). Rather than using a single particle to search for an optimal solution, the whole population is used where the velocities of each member are defined by both a stochastic and deterministic component. While each particle moves randomly, it is partially guided by its own (local) best position as well as the best position of the group (global). The updated velocity vector at time $t+1$ for particle $i$ ($v_{i,t+1}$) is defined as follows:

$$v_{i,t+1} = v_{i,t} + \alpha \epsilon_1 (g^* - x_{i,t}) + \beta \epsilon_2 (x_{i}^* - x_{i,t})$$

where $\epsilon_1$ and $\epsilon_2$ are two randomly initiated vectors with entries ranging between 0 and 1, $\alpha$ and $\beta$ are the global and local learning parameters respectively, $x_i$ is the position of particle $i$ and $g^*$ and $x_i^*$ are the global and local (for particle $i$) best historical location.

Evolutionary algorithms

First proposed by Holland (1992), genetic algorithms (GA) are arguably the most popular variant of the evolutionary algorithm. These methods are a computational model inspired by evolution and the mechanisms of natural selection and are typically deployed as search and optimisation algorithms. The parameters of the user-defined search space are first encoded in the form of chromosomes which can in turn be grouped to form a population. The process begins by initiating a random population representing different nodes in the search space. The fitness (cost) function is then evaluated for each node to determine the fitness value. New search nodes are randomly generated by applying genetic operations on the nodes.
based on their fitness values. This process is repeated until an optimal solution is acquired.

The purpose of the genetic operators is to combine the 'good' structures of each node to produce an improved search node. Common genetic operators are shown in Fig. 4 and include (a) crossover (portions of chromosomes are swapped), (b) reproduction (chromosomes with good fitness values in an old population are preserved in the new population) and (c) mutation (occasional random alteration of a chromosome).

Three alternative evolutionary algorithms include (a) genetic programming (GP), (b) gene expression programming (GEP) and (c) differential evolution (DE). The fundamental difference between these approaches lies primarily in the composition of the individuals within the respective populations. In GAs, individuals are linear chromosome strings of fixed length; in GPs, they are non-linear units with varying shapes and sizes; in GEPs, they are encoded linear strings of fixed length (similar to GA chromosomes) which are subsequently expressed as non-linear units of varying shapes and sizes; and in DEs, they are real vectors rather than binary chromosome strings.

Imperialist competitive algorithm

The imperialist competitive algorithm (ICA) is an alternative evolutionary search and optimisation algorithm proposed by Atashpaz-Gargari and Lucas (2007) and is derived from human being's socio-political evolution. In this case, the initial population is termed 'countries' and is broken into two categories: (a) colony and (b) imperialist. A cost function is used to determine which countries of the initial population are the most ‘powerful’ and are therefore selected as imperialist states. The remaining countries are assigned as colonies of the imperialist states depending on the value of the cost function for each imperialist state. The imperialist state and their respective colonies are denoted an empire. The ensuing optimisation process is described by Fig. 5.
MACHINE LEARNING IN TUNNELLING

Overview

A wide range of ML techniques have been developed for tunnelling applications. Research areas have included TBM automation (Mokhtari and Mooney 2019), tunnel condition assessment (Li et al. 2017, Chen et al. 2019c), anomaly detection (e.g. Yu et al. 2018, Sheil et al. 2020), tunnel profile measurement (e.g. Xue and Zhang 2019), resilience assessment (e.g. Khetwal et al. 2019), structural defect identification (e.g. Ding et al. 2019), tunnel face stability (e.g. Hayashi et al. 2019), rockburst prediction (e.g. Liu and Hou 2019) and intelligent building information modelling (e.g. Zhao et al. 2019b). This review focuses on four tunnelling applications where the use of ML has been most prevalent: (a) TBM performance prediction, (b) tunnel-induced settlement prediction, (c) geological forecasting and (d) cutterhead design optimisation.

TBM performance prediction

A large body of research has focused on the development of improved TBM performance predictions by leveraging recent advances in ML. Table 1 presents an overview of these studies where the corresponding parameters and notation are defined in Fig. 6 (a slurry pressure balance shield machine is shown for illustrative purposes) and Table 2. Research into TBM performance has largely been confined to open mode TBM tunnelling in rock with only a handful of efforts with slurry or earth pressure balance shield TBMs in softer soils (e.g., Mooney et al. 2018, Mokhtari et al. 2020). From Table 1, it is notable that penetration rate (PR) is the most favoured measure of TBM performance, defined as the penetration along the axis of the tunnel per unit tunnelling time (i.e. does not include down times/stoppages). From Table 1, it can be observed that the input parameters are dominated
by ground (rock) properties, with UCS being the most common. It is noteworthy that the selection of input parameters have predominantly been guided by empiricism from previous literature and the application of more robust ‘feature engineering’ techniques, such as principal component analysis (e.g. Salimi et al. 2015, 2016, 2019), in this area has been limited.

Another interesting observation is the insituation of TBM operational (e.g. jacking force (JF), cutterhead torque (T), cutterhead rotation speed (RPM), slurry parameters, etc) and geometric parameters (e.g. tunnel diameter, distance from reception shaft, soil cover etc) as features. This is because many of the training datasets relate to a single construction project and it is a common assumption that TBM and geometric parameters remain constant during a given project and so should not be included in the ML. While this provides good predictability on a case-by-case basis (where one might wish to forecast the performance of the TBM for the current project based on the data gathered thus far), it limits the applicability of these trained ML models to other projects. This is particularly important in the case of ML models as they typically demonstrate a poor ability to extrapolate beyond their calibration space (Ahmed et al. 2010).

The most common ML technique adopted for the prediction of TBM performance is a multilayer feedforward ANN with back propagation. The main difference between the ANN models adopted in the literature is the optimal ANN architecture that was ultimately selected. Even though similar input parameters and datasets have been employed across various studies, the range of architectures that have been adopted is quite wide. For example, Armaghani et al. (2017) and Koopialipoor et al. (2019c) adopted a 7-11-1 \((n_f-n_{h1}-n_o)\) and 5-8-32-8-1 architecture respectively for the prediction of the same dataset (the Pahang-Selangor raw water transfer tunnel (PSWRT)). It is noteworthy that the use of several hidden layers and neurons increases the likelihood of encountering overfitting. Hecht-Nielsen (1987) proved that any continuous function can be represented by a neural network using a single layer with \(n_{h1} = 2n_f + 1\) nodes, albeit using significantly more complex activation functions.
than the conventional sigmoidal functions commonly adopted in the literature. This corresponds to an architecture of 7-15-1 and 5-11-1 respectively for these studies. Other popular ML methods adopted in the literature include fuzzy logic, due to its ability to incorporate empirical evidence/experience and, recently, more flexible and non-linear ML algorithms such as CARTs and RFs. To develop improved methods for the determination of the optimum architecture and the avoidance of local minima, hybrid methods have also been explored by fusing ML models with optimisation algorithms such as ICA (e.g. Naghadehi et al. 2019), PSO (e.g. Armaghani et al. 2018), DE (e.g. Fattahi and Babanouri 2017) and FCM (e.g. Fattahi 2016).

294 Tunnelling-induced settlement prediction

295 Table 3 presents an overview of ML models adopted for the prediction of tunnelling-induced soil settlements, s, as well as tunnel convergence, C. Given the complex nature of tunnelling-induced settlements, the number of features used in these models are notably greater. Furthermore, these features comprise a mix of soil, tunnel geometry and TBM operational parameters. For the studies considered in this review, ANNs appear to have been the ML model of choice pre-2012, although they continue to appear in more recent literature. It is again apparent that a wide range of architectures have been explored from the 47-47-47-47-2 architecture adopted by Kim et al. (2001) to the more compact 3-4-1 architecture proposed by Hasanipanah et al. (2016) and Moghaddasi and Noorian-Bidgoli (2018).

295 While the integration of fuzzy systems has also been used to predict tunnel-induced settlements, the use of SVMs became popular post-2012, quickly followed by more complex and non-parametric methods such as CARTs and RFs. The prominence of these methods for settlement prediction is perhaps explained by the increased complexity of the input-
output mapping process for tunnelling-induced settlements. The datasets used for predicting tunnel-induced settlements are also largely based on a single project rather than multiple projects, with the size of the dataset varying considerably (from 6 to 7650 datapoints).

Geological forecasting

Efforts to predict ahead of the TBM involve identification of geological conditions as well as the size and location of potential obstacles (Schaeffer & Mooney 2016). In these cases, it is desirable to identify changes in soil conditions as shown in Fig. 7. To obtain actionable information during tunnelling, soil conditions must be forecasted sufficiently far in advance of the TBM (typically tens of metres). This is complicated by a deterioration in the accuracy of forecasting techniques with an increase in the forecast horizons.

One approach is to consider the TBM itself as an exploratory tool. A popular implementation of this approach is to first use statistical interpolation techniques (such as kriging) to develop an initial estimate of the ground conditions at the TBM face using available borehole information as shown in Fig. 8 (Gangrade and Mooney 2019, Grasmick et al. 2020). These predictions are subsequently updated using TBM driving data to obtain a more reliable estimate of the ground immediately ahead of the TBM. This methodology has been adopted by Yamamoto et al. (2003) and Sun et al. (2018b). In particular, Sun et al. (2018b) achieved a prediction accuracy of $R^2 = 0.8$ using RFs.

Alternatively, ML can also be used to provide a direct mapping between TBM performance parameters and ground conditions. This approach can be considered the inverse of the techniques reviewed for TBM performance prediction. Liu et al. (2019) used SVR combined with a stacked single-target technique to identify multiple targets from a common dataset, such as UCS, BI, DPW and $\alpha$; this allowed correlation between targets to be incorporated into the prediction model. The driving data used to identify the target variables included
RPM, PR, JF, T and cutterhead power (CP) where a prediction accuracy of $R^2$ between 0.63 and 0.83 was achieved. It is notable that $R^2 = 0.83$ corresponded to the UCS prediction indicating its strong correlation with TBM performance in rock. Zhang et al. (2019c) used SVM, RF and $k$-nearest neighbours (kNNs) to map RPM, T, JF and AR to rock mass type. Zhao et al. (2019a) compared the performance of eight ML models to predict geological type using feature augmentation to improve performance; a traditional ANN was found to provide the best performance. Jung et al. (2019) also used an ANN to predict the ground type from PR, JF and T with an accuracy of $R^2 > 0.9$. The PR parameter was found to be the most influential for predicting ground type, particularly across different sites. Liu et al. (2020) used a hybrid algorithm combining traditional ANNs with simulated annealing to predict rock parameters UCS, BI, DPW and $\alpha$ from RPM, T, JF, PR ($R^2$ between 0.66 and 0.85). Erharter et al. (2019, 2020) used ensemble LSTM networks to classify TBM data into rock behaviour types according to four geological ‘indicators’. Yu and Mooney (2020) employed multinomial logistic regression to characterize the fractional representation of four encountered soil types (sand, clay, silt, till deposits) by an earth pressure balance TBM. The regression model was trained using RPM, AR, chamber pressure, excavated soil mass, thrust force, and 83 boring logs along the alignment. Instead of using TBM operational parameters, Zhuang et al. (2018) used convergence displacements in rock to infer $E_s$ and $v_s$ through inverse analysis. This involved the use of SVR which is optimised using multi-strategy artificial fish swarm algorithm (MAFSA). The MAFSA approach is an ensemble algorithm comprising differential evolution, particle swarm optimization, adaptive step size and phased vision strategy based on artificial fish swarm algorithm (AFSA) to enhance the global search capability and improve convergence speed and optimization accuracy. While numerous geophysical methods have been explored for forecasting geological conditions ahead of the TBM face (e.g. electromagnetic methods, electrical methods, seismic reflection methods, infrared detection methods), very few studies have explored the
The integration of machine learning algorithms to improve geophysical predictions. Both Alimoradi et al. (2008) and Von and Ismail (2017) used an ANN to identify rock characteristics using ground parameters obtained using Tunnel Seismic Prediction (TSP) technology. Although Von and Ismail (2017) reported a prediction accuracy of $R^2 = 0.85$, they noted that the small datasets at the beginning of a project lead to less reliable predictions.

Wei et al. (2018) documented one of the most comprehensive applications of ML to a new ‘Tunnel Look-ahead Imaging Prediction System’ (TULIPS). The TULIPS imaging approach comprises three sets of GPR antennae (low frequency for long-range inspection and two high frequencies to identify small objects) and seismic imaging. The pipeline of their event detection and tracking method is outlined in Fig. 9. An experimental campaign showed that buried obstacles can be successfully identified and tracked using this methodology. Those authors also recommended the development and application of more robust ML models to larger datasets including expert interpretations and ground prediction, TBM and geological exploration data.

Cutterhead design optimisation

The final research area covered by this literature review is the optimisation of the cutterhead design (see Fig. 10) which appear to have focused exclusively on tunnelling in rock. Literature in this area can be further categorised as an optimisation of (a) cutter disc layout and (b) cutter disc geometry. For the cutter layout, the optimisation process has been typically undertaken to (a) minimise eccentric forces (and therefore moments) of the whole system by maximising cutterhead symmetry, (b) maximise excavation efficiency by ensuring adjacent cutters score the tunnel face successively and (c) minimise excavation-induced stress on the cutterhead (e.g. Ji et al. 2016). Other common constraints include (a) cutter discs must remain contained within the cutterhead, (b) cutter discs must not overlap, (c)
cutter discs must not interfere with manholes, ‘buckets’ or joints in the cutterhead, and (e) cutter disc positions should be easily accessible for maintenance (Rostami and Chang 2017).

An example optimisation documented by Huo et al. (2010, 2011) using a multi-objective GA and co-evolutionary GA is presented in Fig. 11. Those authors used three ‘base’ designs as the starting point for the optimisation to reflect current designs used in practice: a multi-spiral (Fig. 11(a)), ‘dynamic star’ (Fig. 11(b)) and stochastic pattern (Fig. 11(c)). Another possible reason for the use of these base designs is that the results of the optimisation process were reported to be highly dependent on the initial cutter pattern. This was also discovered by Qi et al. (2013) using grey rational analysis. Grey rational analysis (GRA) is a form of grey system theory (proposed by Deng (1982)) and solves multiple attribute decision making by combining the entire range of attribute values being considered for each alternative decision into a single value (Kuo et al. 2008). Those authors also found that the polar angle played a more important role on the cutter layout rather than radial distance from the centre point of the cutterhead. Although not discussed in those studies, these findings suggest the occurrence of local optima in these optimisation problems. While multiple alternative optimisation algorithms exist (e.g. grid search, random search), Bayesian optimisation (Brochu et al. 2010) seems suitable for this problem given its robustness to local optima. This is due to its exploration versus exploitation strategy: exploitation initially steers the search process into the direction of the local optima but exploration allows the algorithm to ‘escape’ from the local optimum towards finding an improved global optimum.

On the geometric design of individual cutters, Xia et al. (2012) and Xia et al. (2015) used GA and multi-objective and multi-geologic conditions optimisation (MMCO) to optimise the (a) cutter cutting edge angle, (b) cutting edge width, (c) transition arc radius and (d) caulking ring width between bearings. The optimisation process sought to minimise the cutter bearing load.
This review has identified an increasing trend in the use of ML in the tunnelling space with a significant increase in 2019. It is likely that this trend will persist as advancements in ML continue to be translated into practical domains for routine use and more tunnelling data is shared with the academic community. ANNs have experienced sustained popularity in this area. This is not surprising as ANNs are one of the oldest ML paradigms and are able to capture complex non-linear relationships and generalise within the trained parameter space. The second most popular technique is SVR/SVM. The non-parametric nature of these models means that model complexity remains relatively unaffected by an increase in the number of features and are therefore particularly suited to high-dimensional datasets. This may go some way to explaining their popularity, particularly for settlement predictions due to the larger number of influencing factors. These techniques have been typically coupled with optimisation algorithms to overcome the slow tuning process of the kernel hyperparameters. The use of fuzzy-based methods such as ANFIS and FL in this area stems from their ability to incorporate human experience and their ability to deal with imprecise and noisy data typical of construction monitoring projects. These methods have not experienced the same growth, which is probably due to the increase in ‘big data’ in tunnelling that lends itself to training more robust algorithms. It is also apparent that there has been a significant and recent increase in the use of alternative ML algorithms such as GEP and RF. These models provide a higher level of performance for the sake of model interpretability and can therefore capture highly non-linear trends. The use of probabilistic ML techniques, such as Bayesian networks and Gaussian process regression, for underground construction applications have become more popular in recent years e.g. Zhang et al. (2016), Wang et al. (2017), Chen et al. (2019d), Zhu (2019). These methods are well-conditioned for dealing with noisy and incomplete data typical of a construction site and perform predictions within a principled
CONCLUSIONS AND FUTURE PERSPECTIVES

This paper has presented a comprehensive review of the literature exploring the use of machine learning to inform tunnelling operations. While machine learning has been used to inform a wide range of tunnelling applications, this review has identified four main areas of research, namely TBM performance prediction, tunnelling-induced settlement prediction, geological forecasting and cutterhead design optimisation. Many studies have reported the successful application of machine learning techniques in tunnelling activities with high levels of accuracy. The most popular methods adopted in the literature include artificial neural networks, support vector machines/regression and fuzzy-based methods. A clear trend is evident in the use of ML in tunnelling and this trend is likely to persist as the volume of data produced by modern TBMs continues to grow and the use of machine learning becomes more commonplace. In most instances, investigators have used empiricism (from previous literature) as the basis for the selection of model inputs where the number of features varies considerably across the literature. As the number of parameters captured by modern tunnel boring machines grows, identification of the most appropriate features for training ML models using robust techniques should be central to future research.

Despite its recent advances, machine learning in tunnelling remains a young field with many underexplored research opportunities. Some of these opportunities can be observed by contrasting the methods reviewed in this study with those adopted in other disciplines such as aerospace, healthcare, robotics, and automated vehicles (Mooney et al. 2020). In particular, there is a real need for continued application of machine learning methods employing more principled, probabilistic frameworks such as Bayesian networks and Gaussian process regression. The problems covered by this review appear well-suited to
probabilistic frameworks given the uncertain nature of tunnelling operations and the prevalence of noisy data. This relieves engineers of onerous data pre-processing to denoise large training datasets. Furthermore, probabilistic frameworks provide a robust treatment of overfitting meaning large datasets are not necessarily a prerequisite and deployment of these techniques on a site-specific basis is feasible.

Another important finding of this review is that most of the studies reviewed here have been developed and validated against a single case history. Validation of these algorithms across a broader parameter space are warranted for the industry to gain confidence in these approaches. In addition, the high-risk nature of mistakes in the tunnelling industry means model interpretability is essential for take-up in practice to gain insight into the features driving predictions. It appears essential for the tunnelling industry to begin to consider how best to leverage these recent advances in machine learning to inform tunnelling operations.

REFERENCES


J. C. Dunn (1973) A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, , 3:3, 32-57, DOI: 10.1080/01969727308546046


Kim, C., Khetwal, S., Pei, S., and Gutierrez, M., 2019, September. A Data  


Salimi, A., Moormann, C., Singh, T. and Jain, P., 2015, November. TBM performance prediction in rock tunneling using various artificial intelligence algorithms. In *11th Iranian and 2nd regional tunnelling conference Iran*.


