Aalto University

MS-E2177 - Seminar on Case Studies in Operations Research

Optimal Use of Mortar Systems
Final Report

Niki Leskinen
Tomas Toro
Henrik Sinivuori
Eero Virmavirta
Patrick Linnanen (Project Manager)

June 4, 2024

Contents

1 Introduction

1.1 Background

The Canadian Armed Forces are in the process of improving their use of mortars. The organization responsible for providing technology and advice for the Canadian Armed Forces, Defence Research and Development Canada, is heavily involved in this process to help assure that the mortars acquired are optimal for their needs. The mortar usage is considered in a couple of predetermined combat scenarios that are constructed in collaboration with the client.

Our client for this project is The Finnish Defence Research Agency which is a multidisciplinary research and development organisation that provides advanced research, development, testing, and evaluation services for defence. They are providing consultation for Defence Research and Development Canada to assist in their review of potential mortar acquisitions. The Finnish Defence Research Agency has an artillery ballistics simulation software, EETU, that can help in determine the optimal use of mortars in different combat scenarios. While this software can give accurate results, the simulations take a long time when the scenarios get more complicated so using it to determine the optimal set of actions in a scenario can be very time-consuming.

In this project we develop a model that can provide reliable simulation results in significantly less time than the simulation software. The purpose of creating the model is to speed up the comparison of different sets of actions. The model is then used to determine the optimal use of mortar systems in the given scenarios. We plan to create the model in a format that the client can easily use on their own.

The initial scenarios for the simulations were given by the client. However, because of restrictions in the software and limited time to complete the project, the scenarios are adjusted in order to meet the deadlines. We discuss each adjustment excessively with the client to ensure that the scenarios remain realistic enough.

1.2 Objectives

The objective of the project is to develop a mathematical model to simulate the outcome of a mortar/artillery strike against a near-peer adversary in a predetermined scenario with a limited amount of variables. The model is created using output data from a computationally heavy artillery ballistics simulation software EETU, hence the term metamodel is used of the model developed in this project.

Additionally, the clients expressed interest in several features for the meta-model to answer concerns regarding indirect fire.

1.3 Scenario

In the project topic description and project plan the scenario being modelled was described as in Figure 1. In this scenario, three Armoured Personnel Carriers (abbr. APC) are in a row formation with three squads of infantry in a wider row formation in front of them.

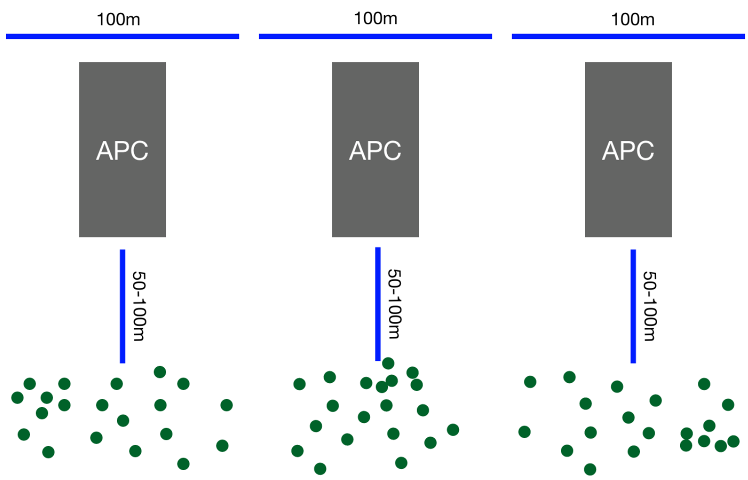

Figure 1: Basic scenario as of the project plan

However, the scenario being modelled was changed after discussions with the client. It was concluded that since the mortars being modelled are only effective on hard targets like APCs on a direct hit or shrapnel impacting on a sensitive section, e.g. a sensor, the APCs would not be initially considered. Removing them from the scenario made it simpler. Additionally, the infantry formation was changed from a straight row to an alternating front-back pattern for both the squads and the whole platoon. This the formation that was used for the rest of the project can be seen in Figures 2 and 3 below.

In the initial scenario various ammunition types and firing platforms were considered, but were later on simplified to only consider two calibers, 81 mm and 120 mm, of fragmentation shells with no regard for the platform.

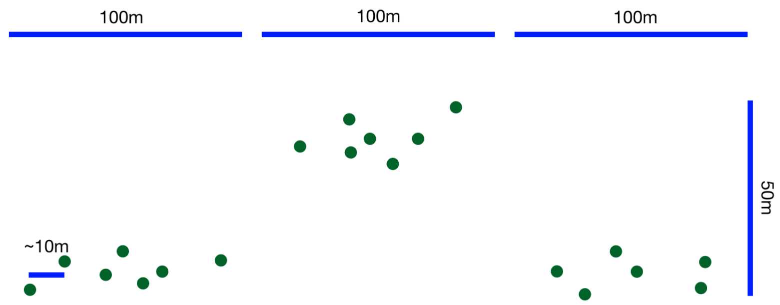

Figure 2: Basic scenario given by the client. Green dots represent soft infantry targets

The Red side also had multiple initial stances and possible reactions; standing, prone or in a foxhole (standing in a hole in the ground, modelled as a smaller box), static or on the move and no reaction or going prone after the first volley. Due to time constraints of the simulating and gathering data, only the initial stances were used.

2 Literature Review

2.1 EETU Literature Review

EETU is an indirect fire simulation software used to simulate casualties on infantry and heavy targets. It was developed for researching complex small scale scenarios as opposed to SANDIS which is higher platoon level simulation tool. It supports different ammunition types from traditional mortar and artillery shells to more specialized air burst and anti-tank shells.

EETU software calculates expected value of casualties based on a distribution depending on several parameters. Adaptive integration is used to improve computational performance by ignoring areas where the individual shell distribution has no impact. The computation remains still relatively heavy for larger scenarios. Artillery strike targeting is simulated with probabilistic distributions. The software supports uniform, precise and multivariate normal distributions for targeting. The projectiles have physics based fragmentation distribution model. The individual shell fragmentation in simulated in three dimensions. This makes it possible to model the impact of shell explosion height on, for example, airburst shell or a tree impact [Rop15]. It also enables studying the effect of uneven terrain. The distribution depends on shell type, projectile terminal velocity, angle of attack and height of the explosion. The kinetic energy of shrapnel is calculated for the hit likelihood distribution with physics based perforation and drag equations to calculate lethal impacts based on the targets armor rating parameter. The in depth operation of the software is discussed in several journals published by the researchers in the Finnish Defence Research Agency [Ber10],[Lap12],[Åke15],

The EETU simulation predicts real life artillery impact accurately with a root mean square error of 1.3%-units. The software results were tested by firing real artillery shells in a test area and calculating the amount of shrapnel hits. The collected data was used to verify and design the program parameters [Ber10],[Ber13]. For a large enough shell count the impact starts to follow a normal distribution due to law of large numbers. A small shell counts have more variance. This is modeled with the distribution used in the software.

The targets follow a scenario with a platoon size enemy in defensive line formation in an open area. The Finnish doctrine is presented in public *Sotilaan käsikirja* (Soldier's handbook) training manual given to military service soldiers[Puo24] (in Finnish). For a Canadian doctrine we discussed with the local expert Major Allen. As a result the defensive formation was designed to be 300 meters wide with around 100 meters depth such that the soldiers are grouped to 3 squads. The squads are in a v-shaped line of 7 men where every other soldier is some meters behind the first line. The squads are in a line with the middle one located deeper behind the 2 side squads. This accomplishes good defensive line of fire coverage while every soldier is not eliminated with a same artillery hit.

The individual target sizes are based on the situation of an individual soldier. The soldier can be standing, prone or in a prepared defensive foxhole. The target size and individual soldier armor are given in EETU. The software is also capable of modeling targets with taking cover behaviour. Then the target begins at for instance standing and moves to prone or cover after first impacts.

In real life scenarios soldiers do not sit still in a foxhole. Targets move before battle and transition to cover when in danger. Therefore the most effective hits are the first ones on not prepared soldiers and the remaining artillery strikes are less efficient due to targets being more difficult to hit in cover. This is taken into account by some modern multiple round simultaneous impact MRSI weapon systems. These fire grenades at different angles and speeds to hit larger amount of grenades simultaneously. While the traditional mortar systems only hit the first volley simultaneously which depends on the amount of mortars in a battery.

2.2 Metamodel Literature Review

Metamodels are models made for modelling another model. In this project the aim was to predict EETU results with a simpler model only depending on model parameters instead of the entire simulation which is computationally heavier. The metamodels are widely discussed in literature [Rop23],[Kan16] and [Bar98].

Fitting a metamodel is done with same methods as fitting any other statistical model. The model prediction accuracy is impacted by having additional layers between the phenomena and the final model. Therefore the original phenomena needs to be difficult or expensive to model directly for a metamodel to be beneficial.

The simplest distribution for estimating an artillery scenario would be a binomial model with each target having same hit likelihood. This would be a reasonable estimate with no terrain differences and the targeting distribution being uniform.

The more complex scenarios require more complex model. The target soldier formation affects the hit likelihood. There are more targets for a shell in some areas increasing the hit likelihood.

The effectiveness of individual shells should decrease after firing more shells because the first shells eliminate targets. This would indicate that there should be cost effective optimal shells fired, after which the value starts to decrease. On the other hand targeting a single target the likelihood of kill increases as we fire more shells

$$\lim_{n\to\infty} 1-(1-p_k)^n = 1,$$

where p_k is the individual shell hit likelihood.

This indicates it is possible to reach 100% casualties by firing enough shells. Military experts have given a target of 20-30% casualties for a unit to be neutralized. This causes the unit to not be combat operational. Seeking to achieve a higher kill rate is likely not cost effective.

The distance from target affects the ballistic trajectory of the projectile. This affects the shrapnel distribution by changing the angle of fall and affects the targeting accuracy.

Overall the EETU simulation results should follow some sort of non linear model. Fitting a metamodel requires testing different model types and using model selection techniques to select the best option for the data. The objective is to have the model available for use after the project some usability and edge cases should be taken into account. Having the model predict casualties for 0 shells fired might for instance lead to confusion on a new end user not familiar with the model.

For traditional models testing linear and polynomial models is a good first step. One option for fitting a model for large data set is to use machine learning tools. Python has several open source libraries. Scikit learn is used for one metamodel [Hac17]. Machine learning solutions can test multiple model types to fit the best fitting model.

3 Data & Methods

3.1 EETU Simulator

The EETU simulator features a graphical user interface (Figure 3) for building and modeling scenarios as well as an API that can be used to integrate the EETU model to other software.

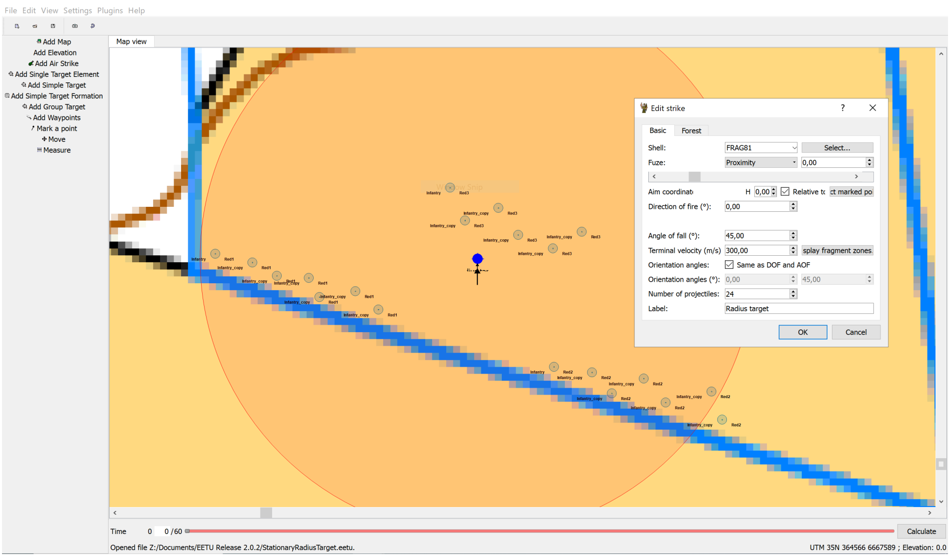

Figure 3: EETU graphical interface

The model takes ammunition, fire mission, target, and terrain parameters as inputs, as presented in Table 1, and outputs the destruction probabilities for target elements as presented in Table 2.

Table 1: Model Inputs

Category	Parameters
Ammunition Parameters	
	Fragment zones
	Fragment mass distribution
	Fragment initial velocity
	High explosive weight
	Fuze detonation height
Fire Mission Parameters	
	Aim point coordinates
	Number of rounds
	Dispersion pattern of rounds
	Projectile angle of fall
	Projectile impact velocity
Target Element Parameters	
	Position
	Vulnerable areas
	Armour thicknesses
	Blast tolerance
	Target behaviour
Digital Elevation Model	
	Terrain height data
	Trees, rocks, and other obstacles

Some commonly used ammunition types were provided in EETU, such as the 81 mm and 120 mm mortar rounds, which were modeled using the ammunition parameters. Similarly, some common target elements were provided, such as soft infantry targets that were modeled as 1m x 1m x 1m cubes.

The model outputs a physics-based probability of destruction for each target element. The model first calculates the probability of a shell detonating at point (x, y) and then it calculates the conditional probability of element is destroyed given the shell detonates in point (x, y).

$$p_{(\text{kill})} = \int\int p_{(\text{impact})}(x, y) p_{(\text{kill} \mid \text{impact})}(x, y) dx dy \tag{1}$$

The integral for combining the probabilities in Equation 1 is computed using an adaptive integration algorithm. The probability of a kill for N rounds is

$$p_{(\text{kill} \mid \text{N shells})} = 1 - (1 - p_{(\text{kill})})^N \tag{2}$$

An example of the data EETU outputs, when firing 36 rounds of 120 mm mortar shells can be seen below in Table 2:

Table 2: Model Output

Target	**Expected Loss**	**Airstrike Type**	**Shots fired**
Infantry1_1	0.22	FRAG120	36
Infantry1_2	0.38	FRAG120	36
Infantry1_3	0.32	FRAG120	36
Infantry1_4	0.35	FRAG120	36
Infantry1_5	0.37	FRAG120	36
...	...	...	...

The "Target" elements in the output data are the targets set up by the user in the simulated scenario, which in this case are infantry targets that are numbered uniquely. Each target has a corresponding "Expected Loss" value, which is the probability the target has been neutralized by the mortar strike. The "Airstrike Type" parameter is used to present the type of ammunition used in the simulation, which in this case was the 120 mm mortar shell. The "Shots fired" parameter expresses the total number of mortar shells launched in the simulation.

3.2 Constructing the Scenarios

The scenarios for the simulations were designed by Defence Research and Development Canada. However, since the scenario creators were not very familiar with the EETU simulation software, we worked together with the client to tweak the scenarios to make them more compatible. One notable adjustment was regarding the firing distance. The scenario descriptions placed great importance on it, but EETU does not have a specific parameter for this. Instead, we had to simulate firing distance using the angle of fall and projectile velocity. There were also some parts of the scenarios that were not formulated explicitly but were required by EETU, such as the formation of troops, the distances between them and the distribution of the mortar fire.

During our meetings with the client, we collaboratively made adjustments to the scenarios with the goal of making them easier to implement in EETU while remaining realistic enough. The reason for tailoring the scenarios for EETU so much was partly due to the time constraints of the course, since without the adjustments we would have had to implement new features to the software to account for all the required aspects. This is something that could be done for future research.

3.3 Data Gathering

The data from EETU was initially harvested manually using the EETU graphical user interface. The group set up a scenario based on the instructions from the client and the target elements' expected losses were simulated with different sets of input parameters determined by the client. Each setup was simulated five times, changing the center of the aim slightly to produce variance between simulations in order to capture the outcome of the simulation more realistically. The output was stored on a spreadsheet with relevant input parameters. An example of the collected data is presented in Table A in the appendix.

However, the graphical interface turned out to be too cumbersome to gather data on sufficiently many permutations of the input parameters, hence the data gathering was automated by creating a script that loops through all the required permutations of different input parameters using the command line interface of EETU, storing the outputs on a spreadsheet.

3.4 Constructing a Metamodel

In developing the metamodel, it is necessary to consider the number and type (continuous, discrete or binary) of input variables. For continuous and discrete variables, the model should be able to interpolate between data points in order to reduce the amount of data needed to gather for constructing the model. Constants, such as the targeting radius, need to be set. The model needs to handle the edge cases, for example firing 0 or 1000 shots, in a sensible way. When 0 shots are fired the expected losses should be 0 and when 1000 shots are fired the expected losses should be almost 100%.

After careful consideration we chose on linear and polynomial regression. We chose linear regression mainly because of its simplicity and multivariate polynomial regression to allow more flexibility. The variable we are trying to explain is `expected_loss` and in the first round of simulations the explaining variables are

- `target_position`
  - The coordinates of the target.
- `ammunition_type`
  - The type of ammunition used (81 mm or 120 mm).
- `number_of_shots`
  - The number of rounds fired to the target area.
- `angle_of_fall`

  - The angle at which the rounds hit the ground.
- target_behaviour.
  - Whether the target is standing, prone or in a foxhole.

The created models are discussed in more detail below.

3.4.1 Linear Regression

Linear regression simply investigates the linear relationships between the explanatory variables and the response variable to find suitable coefficients for the explanatory variables to predict the values of the response variable. [Fre09]

In this scenario, a linear model is developed to predict the value of expected losses based on the explanatory variables presented Section 3.4. The model investigates the mean expected losses of the whole squad, rather than the expected losses of each target separately. We made this decision to allow the model to provide more accurate results, as the discretely modeled target position could have unrealistic effect on the expected losses. This means that the variable `target_position` is not considered when developing the linear regression model.

The fitted coefficients of the linear model are in Table 3. The coefficients are fitted using function `fitlm` from the Statistics and Machine Learning Toolbox in Matlab version R2024a. We use 80% of the datapoints for fitting the model and the rest for testing the accuracy. The mean squared error of the test points was 4.0895×10^{-4}, meaning that the model predicted the outcome of the test points accurately on average. However, due to the linearity of the model, the handling of edge cases cannot be implemented into the model without also largely deteriorating the predicting abilities within the range of the training data.

Table 3: Coefficients of Multiple Linear Regression Model

Variable	**Coefficient**
`Intercept`	-0.51421
`ammunition_type`	0.1809
`number_of_shots`	0.0092419
`angle_of_fall`	0.0067835
`target_behaviour`	-0.11477

3.4.2 Multivariate Polynomial Regression

In multivariate polynomial regression, the relationships between the input variables and target variable are allowed to be nonlinear. For example, a bivariate linear model $y = a_0 + a_1x_1 + a_2x_2$ can be extended to a second-degree nonlinear model $y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2$. Multivariate polynomial regression allows for more flexibility than linear regression but is prone to multicollinearity, i.e., the variables being interdependent on each other. [Sin13]

When choosing the degree of the polynomial, we consider the mean squared errors (MSEs) of models with different degrees. The range of degrees in consideration is $[1, 9]$ because for degrees higher than that the computation times become very long. The MSEs of models with the different degrees are given in Table 4. We can see that with degree 7 we get the lowest MSE but also that with degrees from 4 to 9 the MSEs are very similar. Higher degree polynomials can be unreliable in extrapolation so we choose degree 4 for our model. The improvements on local fit by choosing a higher degree would probably be minimal.

Table 4: Degree Selection

Degree	**MSE**
1	0.00904
2	0.00262
3	0.00126
4	0.00085
5	0.00075
6	0.00068
7	0.00066
8	0.00068
9	0.00069

For implementing the model we use tools from the popular open-source machine learning Python library `scikit-learn`, more specifically the classes `PolynomialFeatures` and `LinearRegression` [Hac17]. The features and coefficients of the resulting polynomial are given in Table 5. The total number of features can be calculated with $\binom{n+d}{d}$, where n is the number of original features and d is the degree of the polynomial. We have five features {`target_position`, `ammunition_type`, `number_of_shots`, `angle_of_fall`, `target_behaviour`} and our polynomial is of degree five so the total number of features is $\binom{5+4}{4} = 126$. Unlike the linear regression model, this model gave significantly more accurate results when we included the variable `target_position` in developing the model.

Table 5: Coefficients of Polynomial Regression Model

Feature	**Coefficient**
`Intercept`	2.959659×10^{-11}
`target_position`	3.422635×10^{-8}
`ammunition_type`	-8.046800×10^{-9}
`number_of_shots`	2.224997×10^{-9}
`angle_of_fall`	5.492641×10^{-15}
$\vdots$	$\vdots$
$\texttt{angle\_of\_fall}^4$	$-3.531098 \times 10^{-10}$
$\texttt{angle\_of\_fall}^3 \times \texttt{target\_behaviour}$	-9.305780×10^{-8}
$\texttt{angle\_of\_fall}^2 \times \texttt{target\_behaviour}^2$	2.202249×10^{-6}
$\texttt{angle\_of\_fall} \times \texttt{target\_behaviour}^3$	7.487199×10^{-8}
$\texttt{position}^4$	1.968918×10^{-9}

We use 80% of the data for training the model and the remaining 20% for testing it. The trained model is then exported to a file so that it does not have to be retrained every time it is used.

4 Results

To test the created models, we created a set of test scenarios. The test scenarios are designed so that we can use them to review the models' capabilities of giving accurate results when we vary the different parameters. We use combinations of parameter values present in the training data and values not present in the training data to have meaningful comparisons. For example, we use the `angle_of_fall` values of 83, 87, 43 and 47 in the tests while we used the values 45 and 85 in the training data. In reality, the angle at which the projectile hits the ground can be pretty accurately determined beforehand so there is no need for large variation in it.

In Table 6 we have the expected losses given by EETU and the two metamodels. The scenarios are given as sets {`ammunition_type`, `number_of_shots`, `angle_of_fall`, `target_behaviour`}. We compare the expected losses given by the linear and polynomial models to the values given by EETU. The expected losses are an estimate of inflicted casualties to the 20 soft targets on a scale of $[0, 1]$.

Table 6: The average expected losses for the different models in the test scenarios.

Scenario				EETU	Linear	Polynomial
{FRAG120,	12,	85,	standing}	0.3226	0.4203	0.3058
{FRAG81,	18,	85,	foxhole}	0.1831	0.1801	0.1835
{FRAG120,	42,	45,	standing}	0.4019	0.4262	0.3825
{FRAG81,	48,	45,	foxhole}	0.1843	0.1860	0.2272
{FRAG120,	24,	87,	standing}	0.5399	0.5448	0.5399
{FRAG81,	36,	83,	foxhole}	0.3317	0.3329	0.3234
{FRAG120,	42,	87,	foxhole}	0.5945	0.5964	0.5924
{FRAG81,	48,	83,	standing}	0.5606	0.5586	0.5318
{FRAG120,	42,	47,	foxhole}	0.3046	0.3250	0.3002
{FRAG81,	48,	43,	standing}	0.2572	0.2872	0.2699

We examined the expected losses as a function of number of shots fired. The expected losses as a function of number of shots when the targets are standing and the angle of fall is 85 degrees for the different models are in Figures 4, 5, 6 and 7. With both models, we can clearly see that with the 120 mm mortar shell the expected losses are much higher than with the 81 mm mortar shell.

In the linear graphs we see that the edge cases, especially in the lower end, can not be handled. From Table 6, we see that apart from the first row the linear model gives quite similar values than EETU but that is at the cost of the model predicting significant losses even when no shots are fired. Also the linear increase in the expected losses may not be realistic because when the losses increase, there are fewer targets left so the rate should decline when the number of shots increases. That being said, the linear model gives a rather accurate approximation of EETU results when the number of shots is near the range used in the training data, which is 24 to 36 shots fired. The further from that range we move, the more unrealistic the results become.

In the polynomial graphs we can see that the rate of increase in the expected losses decreases as the number of shots increases as it would in real life. We can also see that the zero edge case is handled correctly. In Table 6, we can see that the values given by the multivariate polynomial regression model are similar to the values given by EETU in all of the test scenarios. There are some differences, most notably with the scenario $\{\text{FRAG81}, 48, 45, \text{foxhole}\}$, so the model is likely able to approximate values from some regions better than from others.

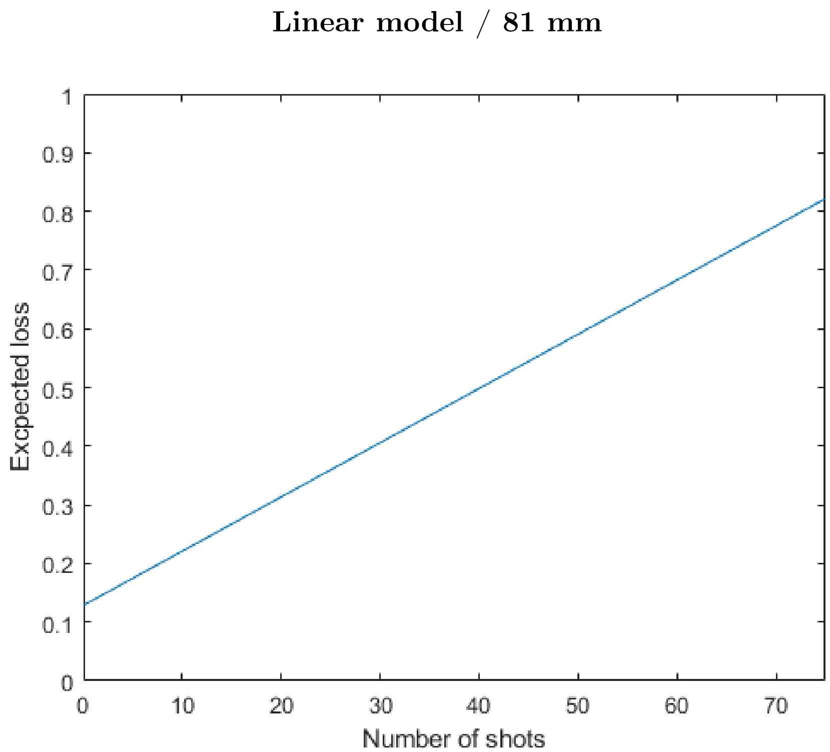

Figure 4: Average expected loss as a function of number of shots for the 81 mm mortar shell given by the linear model.

Linear model / 120 mm

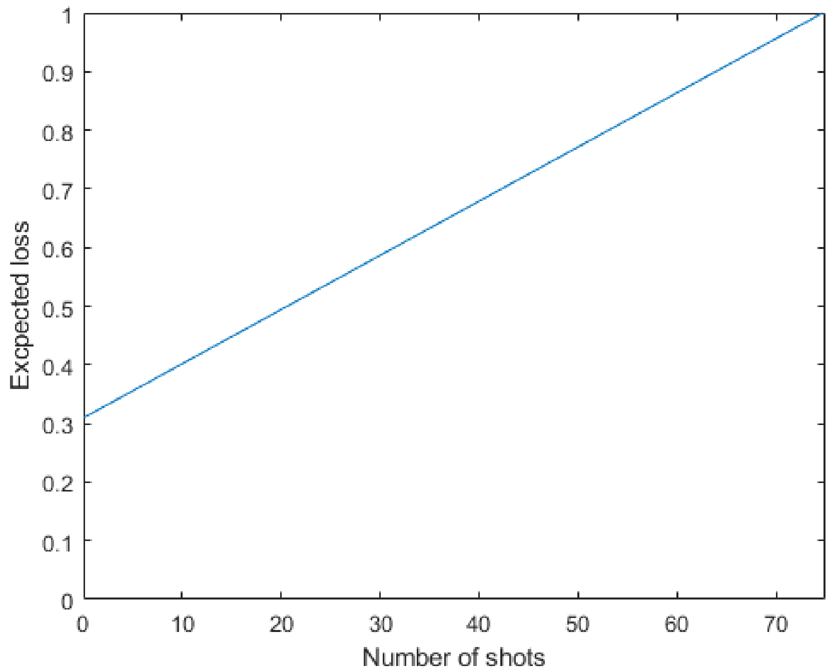

Figure 5: Average expected loss as a function of number of shots for the 120 mm mortar shell given by the linear model.

Polynomial model / 81 mm

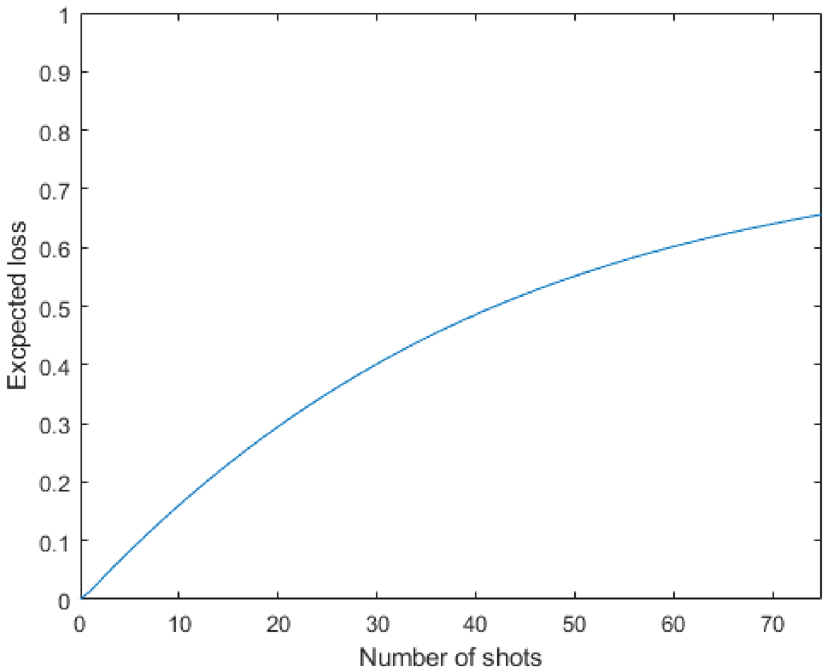

Figure 6: Average expected loss as a function of number of shots for the 81 mm mortar shell given by the polynomial model.

Polynomial model / 120 mm

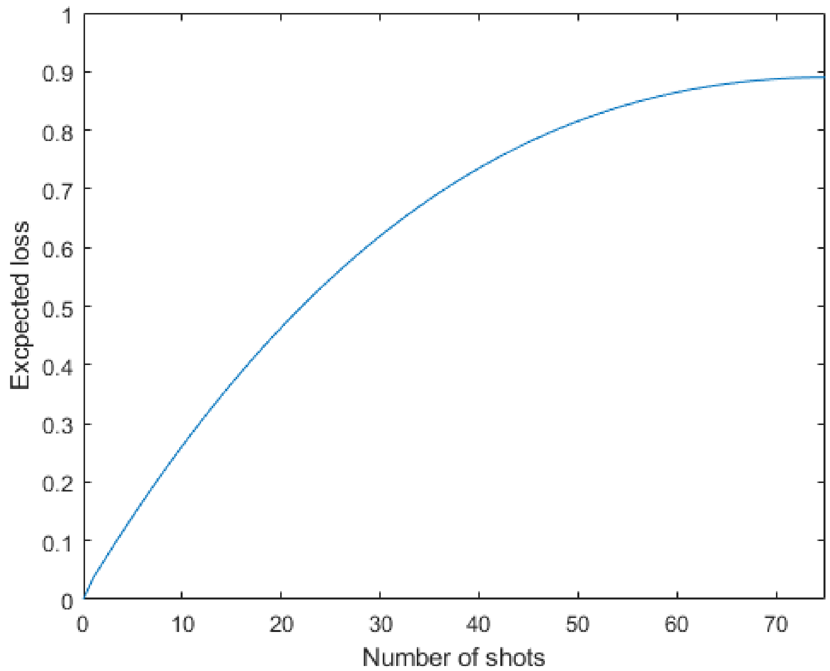

Figure 7: Average expected loss as a function of number of shots for the 120 mm mortar shell given by the polynomial model.

5 Discussion

The objective of the project was to develop a metamodel for EETU scenarios. This has not been done before and the first objective of the project was the overall feasibility analysis. The second objective was then to develop the best possible model for increasingly complex scenarios with the time constraints of the course.

Based on the results the development of a metamodel is possible. The first data set based metamodel gives decent results for the limited amount of data. This indicates that a better model should be doable with more data.

Unfortunately, gathering more data efficiently turned out to be a very time-consuming effort. An expert provided basis of a script with which to gather data programmatically, but due to EETU simulation program constraints and lack of time to debug the code, no useful data was gathered. Doing so is possible, but was not feasible in the constraints of the project.

We identified some challenges in the metamodelling process. Developing a good model requires plenty of data. The first data collection phase was done manually which was slow and labour intensive. This method is not recommended for further development of metamodels. We started testing more automatic data collection script with ability to edit the scenario file and run it from the command line in the last weeks of the project. This process is significantly faster than the manual data gathering. The EETU runtime still remains a limit for the automatic method. This could be alleviated by dividing the data to suitable batches and running them automatically would be one solution. Memory usage would have to be monitored to not run out of program memory. If this works it is a good method of generating large enough amount of data for a larger metamodel.There is still manual work involved in preparing the scenarios. Editing the target positions and quantities might be too complex to automate.

The metamodel prediction accuracy was analyzed in the results chapter. Metamodel polynomial version is functioning as expected. For both 81mm and 120mm the expected casualties as a function of shells fired increases non linearly as seen in figures 7 and 6. The linear model has linear behaviour as expected but the accuracy is also decent outside of the edge cases. The optimal use of shells is theoretically the least amount of shells required to inflict desired casualty rate for instance the 30%, as the additional shells have worse efficiency. The shell use can be optimized by taking the smallest amount of predicted shells to reach the required casualties. This is around 24 81mm shells and 12 120mm for 30% casualties based on the polynomial model. This indicates that a 120mm mortar needs to fire less for a similar impact. This is to be expected from a bigger mortar round. The cost factor of shells is less clear 81mm shells are easier to transport and cheaper. There are in addition, other practical factors influencing the decision. The 81mm mortar is more mobile while the 120mm mortar can support from longer distance possibly supporting more friendly units in the front line with a smaller amount of batteries. Both weapon systems are in use in Finnish military indicating which is more useful to be situational.

5.1 Follow-up studies

The more advanced scenarios with moving targets and vehicles were left out of the current metamodel due to time constraints. The vehicles are not the primary target for a mortar system which specializes in mobile supporting indirect fire against soft infantry targets. Additional scenarios should be interesting topics for further development. The modelling process used in this report works for these as well.

A metamodel predicting multiple scenarios needs to be trained individually for each scenario. This is done by modeling the scenario in EETU, collecting data from the scenarios and fitting a metamodel. The larger metamodel then selects from the trained submodels. A single model predicting multiple different scenarios is unlikely to work as reliably due to fast increasing amount of parameters and the differences in the target size and behaviour.

Additional interesting research topic might be to create a more accurate metamodel with model stacking. It would be possible to create a metamodel with a porftolio of models with different weights. This would likely improve edge case accuracy since it would be possible to combine a good edge case handling model to a more accurate model which is worse in the wider data range.

Anti-vehicle effectiveness model might be interesting to develop for artillery or drone based weapon systems. Considering the Ukraine war with the development of new autonomous drone weapon technologies a prediction for estimated amount of drones needed might be an useful tool in Eetu. While EETU is an impressive software for simulating, we identified some useful quality of life features for the future development. Implementing some missing Microsoft Windows key shortcuts such as ctrl+c ctrl+v would make editing larger formations faster. The premade target configurations could have some new options implemented. A basic v shape line defensive formation would be fairly simple to implement based on the already existing square formation.

Bayesian models might be an interesting alternative to the metamodels discussed in this report. Since the metamodel(s) would be used to time-efficiently iterate different parameter combinations before running a more computationally intensive simulations, a Bayesian simulation optimisation approach could be an effective alternative to reduce the number of intensive simulation runs. Also, some sort of Bayesian or Gaussian linear model might work for predicting the casualties. This would have the additional benefit of producing the full distribution instead of the expected casualty giving more information on the likelihood achieving the desired casualties with the amount of projectiles. The metamodel compared to EETU loses the information of the full distribution in this reports metamodels.

6 Conclusions

The objective of this project was to research the possibility of utilising a metamodelling approach to ease the computational load and time required for indirect fire simulations, construct a metamodel if it is possible and demonstrate its ability to model the simulation model and give a heuristic for casualties given parameter variations in a specific scenario. It can be concluded that the project has succeeded, although not to the team's satisfaction. A metamodel has been implemented, it is considerably more time-efficient than EETU and gives somewhat accurate approximates for casualties in the given scenario outside of the parameter values used in its training.

However, various avenues of improvement are available even though the project is over. Given more time, significantly more data could be gathered and more advanced simulations constructed, which could include the initially desired armoured personnel carriers, reactions from targets, and so on. In addition to more parameters, a larger volume of data - most likely gathered with an automated script - would enable investigating more complex models without danger of overfitting. The team also discovered multiple points of improvement for the simulation software and came up with other approaches to solve the problem, even though there was no time to pursue them due to time constraints.

7 Self Assessment

7.1 Execution of the Project Plan

The initial plan was the following sequence of tasks:

1. Get familiar with the EETU software and theory
2. Simulate scenarios with EETU
3. Research suitable metamodel types for the simulation data
4. Fit metamodels for simulation data

5. Investigate more complex metamodels with additional scenario parameters

As already pointed out in the initial project plan, it was difficult to gauge the time to run the simulations and fit the metamodel. However, it also turned out that it was difficult to estimate how much time it would take to get familiar with the EETU software and the theory.

Hence, instead of progressing linearly from one task to the next, we started on task 1, then proceeded to task 2 while still working on task 1, and then proceeded to task 3 while continuing to work on tasks 1 and 2 in parallel, and so on. The iterative nature of the project was already acknowledged in the project plan, but instead of iterating through tasks 2-5, we iterated through tasks 1-4 while also working on all the tasks simultaneously. The task number 5, investigation of more complex metamodels with additional scenario parameters, was left for future research.

Diverting from the initial plan was mostly due to challenges in the use of EETU and understanding the theory behind mortar fire, as well as due to the time-consuming nature of simulating the scenarios and collecting the data.

The first batch of data was collected by late March 2024 the research on how to model that data started in early May 2024, while still running simulations and collecting more data. In mid-May 2024 the final batch of data was collected, which was used to determine the metamodels.

7.2 Scope of the Project

The project's scope was narrowed due to time constraints. Initially, the objective was to investigate more complex metamodels with more parameters to provide flexibility in simulating different scenarios. However, the number of parameters was reduced because each additional parameter would have at least doubled the number of unique combinations of the parameters. Each unique combination needed to be simulated at least five times to gather sufficient data. Therefore, including all the initially planned parameters in the metamodel would have resulted in too many scenarios to simulate.

Additionally, creating an optimization model that utilizes the metamodel in order to determine the optimal artillery setup was left out of the project's scope to focus the resources on gathering data and the creation of the metamodel.

7.3 Tools used in the Project

A significant portion of the project was spent with the EETU simulator, which was very daunting to learn and "clunky" to operate. On the project team side, we could have reached out and asked for technical help sooner rather than later, but on the other hand, the client could have provided resources to get started with the software. Utilising the simulator is challenging and frustrating without comprehensive documentation. For example, data gathering with a script was impossible to figure out without reaching out to the writer of the program, and debugging the script even more so. The learning to efficiently and correctly use the simulator was a great drain on time resources.

7.4 Communication with the Client

The project team held meetings every other week with the the entire clientele consisting of the representatives from both the Canadian Army and the Finnish Defence Forces to communicate the progress of the project and to discuss any obstacles the team might have come across during the project. The meetings were informative and productive, and helped the team gain domain-specific understanding, and narrow down the scope when needed.

Communication with the client was occasionally slow, which made determining the correct domain-related input parameters time-consuming, which then delayed the collection of the data. Additionally, significant time was devoted to integrating the requested parameters into the EETU software and navigating its command line interface due to a limited understanding of how to use the software. Therefore, in hindsight, a comprehensive training session in the beginning of the project on the basics of mortar fire, along with guidance on using EETU and utilizing its pre-modeled elements, would have given the team a significant head start.

7.5 General Discussion

The biggest setback of the project was the time management regarding the gathering of the data. Since the data was gathered later in the project than initially planned, the time the team had to assess different options for fitting the metamodel was limited.

However, despite not achieving all goals initially set by the client, the project has laid a firm foundation to allow future researchers to gather more data and obtain more accurate and flexible metamodels that model the scenario in question.

8 References

References

[Bar98] Russell R Barton. "Simulation metamodels". *Proceedings of the 1998 Winter Simulation Conference. (Cat. No. 98CH36274)*. Vol. 1. IEEE. 1998, pp. 167–174.

[Fre09] David A Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2009, pp. 26–27.

[Ber10] Esa Lappi Bernt Åkeson. "An improved model for simulating indirect fire" (2010), p. 1.

[Lap12] Esa Lappi. "Computational methods for tactical simulations". *Maanpuolustuskorkeakoulu taktiikan laitos* Julkaisusarja 1.1/2012 (2012), 1–TODO.

[Ber13] Ville Petterson Bernt Åkeson Esa Lappi. "Validating indirect fire models with field experiments". *Journal of Defence Modeling and Simulation* 10.4 (2013), pp. 425–434.

[Sin13] Priyanka Sinha. "Multivariate polynomial regression in data mining: methodology, problems and solutions". *International Journal of Scientific Engineering Research* 4.12 (2013), pp. 962–965.

[Åke15] Bernt Åkeson. "Physics-based lethality model for fragmenting ammunition". *Journal of Defence Modeling and Simulation* (2015).

[Rop15] Juho Roponen. "Master's thesis Simulating artillery fire in forest environment" (2015).

[Kan16] Jussi Kangaspunta. "Resource allocation methods for defence and infrastructure systems" (2016).

[Hac17] Gavin Hackeling. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017.

[Rop23] Juho Roponen. "Computational models for adversarial risk analysis and probabilistic scenario planning" (2023).

[Puo24] Puolustusvoimat. *Sotilaan käsikirja*. 2024.

A Appendix – Sample of the Training Data

target	squad	expected_loss	ammunition_type	shots	angle_of_fall	terminal_velocity	target_position	target_radius	field
Infantry_1_1	Red1	0.083	FRAG81	24	45	300	standing	159	open
Infantry_1_2	Red1	0.153	FRAG81	24	45	300	standing	159	open
Infantry_1_3	Red1	0.122	FRAG81	24	45	300	standing	159	open
Infantry_1_4	Red1	0.145	FRAG81	24	45	300	standing	159	open
Infantry_1_5	Red1	0.150	FRAG81	24	45	300	standing	159	open
Infantry_1_6	Red1	0.153	FRAG81	24	45	300	standing	159	open
Infantry_1_7	Red1	0.151	FRAG81	24	45	300	standing	159	open
Infantry_2_1	Red2	0.153	FRAG81	24	45	300	standing	159	open

Table 7: Sample of the metamodel training data