Evaluating the Effectiveness and Usability of Microsoft Threat Modelling Tool in Undergraduate Cybersecurity Education

Nor Laily Hashim, Ahmad Zuhairi Bin Mohd Yusri
School of Computing, Universiti Utara Malaysia, UUM Sintok, Kedah, Malaysia

***Abstract*—As cyber threats evolve, equipping students with hands-on experience in identifying and mitigating system vulnerabilities is critical for developing a cybersecurity-aware workforce. There are a variety of threat modelling tools available on the market, and it is challenging for educators to select the best tool for their students to learn and identify any possible threats that may exploit system vulnerabilities. This study investigates the effectiveness and usability of the Microsoft Threat Modelling Tool (MTMT) among undergraduate students, addressing the need for a practical tool in cybersecurity education. This study was conducted in four phases. The first phase involves conducting a comprehensive literature review to understand the features, capabilities, and tools of the threat modelling tools being compared, specifically the MTMT. Phase two consists of defining the evaluation criteria for assessing the tool's effectiveness and usability. Criteria for error frequency, ease of use, and user-friendliness will be developed, with particular focus on their relevance to educational environments, especially for undergraduate students. Phase three involved data collection, during which participants were recruited and had hands-on sessions with the tool. Training sessions were conducted using case studies to familiarise participants with the tool's features and functionalities. The last phase involves developing assessments to evaluate participants' knowledge, effectiveness and usability of the tools. The evaluation includes structured usability testing and post-assessment of students' knowledge and skill acquisition. Findings reveal that MTMT enhances students' comprehension of threat modelling concepts, bridging the gap between theoretical knowledge and real-world cybersecurity practices. However, the study also highlights areas for improvement in the tool's interface and documentation to better support student learning. These insights enhance educational strategies, foster active learning, and equip students for real-world cybersecurity challenges. The results emphasise the tool's potential to strengthen the integration of threat modelling into the cybersecurity field, thereby fostering essential skills for safeguarding organisational and digital infrastructures. The novelty of this study lies in the methodology used to measure the effectiveness and usability of the threat modelling tool. The tool's effectiveness was measured using the effectiveness formulas from ISO/IEC 25022:2016(E), while its usability was measured using the System Usability Scale (SUS).**

***Keywords*—Cybersecurity education; threat modelling; stride; usability testing**

I. Introduction

The increasing complexity of cybersecurity threats requires the practical training of future professionals in this field. One vital component of this training is the practice of threat modelling tools, such as the Microsoft Threat Modelling Tool (MTMT). MTMT is a commonly used tool for systematically analysing system vulnerabilities, creating data flow diagrams, and proposing security mitigations based on identified threats. Threat modelling is an essential technique for identifying and understanding potential threats, and for suggesting and prioritising mitigations to safeguard valuable system assets. This tool is primarily designed to assist developers and cybersecurity teams in mitigating identified potential security vulnerabilities early in the software development lifecycle. By integrating MTMT into undergraduate cybersecurity curricula, educational institutions can enhance students' understanding of security principles and practices in the digital world, ultimately preparing them for real-world cybersecurity challenges.

The variety of threat modelling tools available on the market can be challenging, and organisations may need help selecting the best solution for their unique requirements. By undertaking a thorough review of available tools, organisations can gain insights into their strengths and shortcomings, enabling them to choose the best tool for their situation. The study by reference [19] used different evaluation criteria to measure practical threat modelling tools. A study evaluating the effectiveness of Threat modelling tools among undergraduates [12] was similar to this study. This study, however, does not focus on assessing usability or on using other measures of the tool's efficiency.

Evaluating the effectiveness and usability of MTMT among undergraduate students is crucial for several reasons. This assessment allows educators to gain valuable insights into how well students implement and utilize the tool, enabling them to highlight areas of strength and weakness in supporting learning outcomes.

By focusing on the student's perspective, the evaluation ensures the tool's design is truly user-centred, fostering an interface that encourages engagement and facilitates effective learning. Furthermore, the feedback gathered through this process is instrumental to the tool's iterative development, enabling it to improve and adapt to the ever-changing landscape of educational needs and technological progress. This comprehensive evaluation approach not only enhances the immediate learning experience but also contributes to the long-term improvement of educational technology. As students engage with MTMT, they not only learn to identify threats but also develop the critical thinking skills necessary to evaluate and implement security measures, thereby fostering a more robust cybersecurity education that aligns with industry needs.

The structure of this research study follows a logical progression, beginning with a literature review section that provides a theoretical foundation and contextualizes the study within existing threat modelling tools and concepts. Following this, the methodology section outlines the research approach, detailing four distinct methods employed to address the research objectives. The study then describes the data sources used, explains the collection procedures during the experiment, and outlines the analytical techniques employed to ensure robust, reliable findings. The results section presents the empirical findings from the data analysis, providing a clear overview of the study's outcomes followed by a discussion section. The study concludes with a few key findings, discusses their implications, addresses the research questions, and suggests potential improvements for future research in cybersecurity studies.

II. Literature Review

Threat modelling is a critical practice in software development that emphasises the proactive identification and mitigation of security threats throughout the Software Development Lifecycle (SDLC). By integrating threat modelling early in the design phase, developers can significantly reduce the costs associated with implementing security measures post-release, which can be up to thirty times higher than addressing these issues during initial design [1]. The process involves a systematic approach to understanding potential vulnerabilities and threats, allowing security analysts to evaluate system architecture effectively [2]. As highlighted by [3] and further supported by [4], threat modelling not only enhances the security posture of applications but also fosters a culture of continuous improvement in security practices. This structured methodology enables developers to make informed decisions about implementing security features tailored to the specific context of their systems, ultimately contributing to more resilient software solutions [5][6].

Data Flow Diagrams (DFDs) are essential tools in system analysis and design, providing a visual representation of how data moves through a system. They help decompose complex systems into manageable components, enabling analysts to understand the flow of information among processes, data stores, and external entities. DFDs facilitate communication between technical and non-technical stakeholders by providing a clear, intuitive graphical depiction of data interactions, which is particularly beneficial for identifying inefficiencies and potential security vulnerabilities [5]. However, creating effective DFDs requires a thorough understanding of the system's architecture and careful attention to detail to avoid oversimplification or misrepresentation of data flows [7]. Despite their limitations, such as the inability to capture timing and sequencing information, DFDs remain a powerful method for analysing systems and guiding the threat modelling process by illustrating how data traverses through various components.

The STRIDE methodology, developed by Microsoft, is a widely recognised framework for threat modelling that categorises potential security threats into six distinct types: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. This systematic approach allows security analysts to effectively identify vulnerabilities in software systems and devise appropriate mitigations [8] [9]. By employing STRIDE in conjunction with DFDs, analysts can visualise data interactions within an application, thereby enhancing their understanding of how threats may manifest at various points in the system [10]. The methodology not only aids in recognising design flaws but also aligns with established secure software processes, such as OWASP's Comprehensive Lightweight Application Security Process (CLASP) and Microsoft's Security Development Lifecycle (SDL), thereby reinforcing its credibility and utility in fostering secure application development [11]. Overall, STRIDE serves as a foundational tool for organisations aiming to address security risks throughout the software development lifecycle proactively.

Measuring the effectiveness of threat modelling tools, particularly the MTMT, is essential for understanding their impact on identifying and mitigating security threats in software development. Research indicates that integrating threat modelling into the software development lifecycle significantly enhances the ability to detect vulnerabilities early in the design phase. For instance, a study involving undergraduate students demonstrated that those using MTMT identified a broader range of threats than those using traditional manual threat modelling methods. The students reported that the tool facilitated a more structured approach to threat identification, allowing them to categorize threats effectively using the STRIDE methodology [12]. Furthermore, feedback highlighted that MTMT not only increased the number of identified threats but also provided valuable insights into potential mitigation strategies, thereby reinforcing its role as an effective educational tool for teaching secure software engineering practices [13]. Overall, evaluating the effectiveness of such tools is crucial for refining their functionalities and enhancing user understanding, especially at the educational level of understanding security risks in software applications.

Usability testing plays a crucial role in evaluating the effectiveness of tools like the MTMT, particularly among undergraduate students who may lack prior experience in software security. A study examining the use of MTMT in a secure software engineering course revealed that usability testing methodologies, such as moderated and unmoderated testing, were instrumental in assessing students' ability to navigate the tool and identify potential security threats. The findings indicated that students performed better with MTMT than with manual threat modelling methods, highlighting the tool's user-friendly interface and guided assistance features. Student feedback highlighted specific usability challenges, such as understanding DFDs and STRIDE techniques, which are essential to effective threat modelling. This suggests that incorporating usability testing into the educational framework can significantly enhance students' learning experiences and their ability to utilise security tools effectively [12].

The existing works related to this study are by [14], which investigates the added value of specific threat modelling applications, namely the MTMT 2016 and the Tutamen tool. The study evaluates these tools based on accuracy in automatic threat generation, time efficiency in executing analyses, and user-friendliness, highlighting their impact on enhancing security processes. The study proposes using a combination of case studies, expert evaluations, and tool demonstrations to

assess the effectiveness of threat modelling tools among undergraduate students.

In [15], the authors focus on assessing various threat modelling frameworks, examining eighteen methodologies, including STRIDE, PASTA, and OCTAVE. This study synthesises their strengths and weaknesses, offering insights into their comprehensiveness and applicability for effective threat modelling. Overall, the research aimed to understand the current landscape of threat modelling methodologies and frameworks, evaluate their suitability for different use cases, and provide practical insights into implementing threat modelling in organisations.

In [16], the authors present a comprehensive taxonomy and qualitative evaluation of several threat modelling tools, including MTMT, OWASP Threat Dragon, and IriusRisk. The study compares these tools based on criteria such as model form, threat libraries, evaluation methods, and SDLC integration, providing a structured framework for understanding their functionalities. The study also discusses the limitations of existing threat modelling tools and proposes directions for future research. Overall, the taxonomy presented in the study provides a valuable framework for understanding and categorising threat modelling tools, helping cybersecurity professionals make informed decisions about tool selection and implementation.

III. Methodology

A. Phase 1: Experimental Design

This study involved conducting a comprehensive literature review to understand the features, capabilities, and tools of threat modelling. This phase aimed to identify the key features, benefits, and weaknesses of MTMT. By thoroughly examining the available literature, a detailed analysis of MTMT's strengths and limitations was produced.

B. Phase 2: Experimental Setup

In this phase, students learned to use the MTMT through a combination of structured learning materials and self-paced exploration. The experimental setup was designed to provide a flexible learning environment, ensuring students had the resources needed to understand and apply the tool effectively. The experiment was conducted in a remote, self-paced learning format, allowing students to engage with the materials and complete the tasks at their convenience.

Students began their learning journey with video-based instruction. They were provided with a lecture video introducing the core features of the MTMT tool, its importance in security threat analysis, and instructions for downloading and installing it. Following this, two hands-on demo videos were made available, offering a step-by-step walkthrough of the tool's core functionalities and how to solve the case study. These included creating DFDs, proposing mitigations, and generating a comprehensive threat model report. These videos ensured that students had a clear, practical understanding of the tool's application. Students engaged in a hands-on, realistic case study focused on analysing a realistic software system architecture. There are two main tasks in this case study.

Task 1 requires students to create a DFD to visually represent the flow of data among the key components of a given case study of a company's Single Sign-On (SSO) system. This task aimed to identify critical paths, trust boundaries, and interactions within the system. The expected outcomes for this task include developing a comprehensive DFD that accurately illustrates the key entities and data flows of the SSO system, and identifying trust boundaries and critical interactions where potential vulnerabilities may exist.

Task 2 involves students proposing security mitigation strategies. The objective here is to identify and recommend effective security measures to address vulnerabilities detected in the SSO system, utilising the MTMT tool's automated threat identification feature. Expected outcomes for this task include a well-defined list of mitigation strategies explicitly tailored to the identified threats within the SSO system, along with an enhanced understanding of how the MTMT aids in prioritising security issues and proposing structured solutions.

Building on insights from the instructional videos, participants utilised the provided guidelines, which detailed system specifications, data flows, and trust boundaries. This structured approach enabled students to systematically apply the MTMT tool to assess the architecture's integrity and security. By simulating real-world scenarios, the experiment not only enhanced their understanding of software design principles but also fostered critical thinking and problem-solving skills essential for navigating complex systems in professional environments.

C. Phase 3: Data Collection

The data collection phase involved gathering feedback on the effectiveness, based on the overall score achieved by all students in their threat model submissions and the MTMT's usability evaluation. An online form was used to collect both completed threat model submissions and detailed feedback from participants regarding their experience using the tool. The feedback form was designed to collect both quantitative and qualitative data. For quantitative data, participants were asked to rate various aspects of their experience on a Likert scale, including the MTMT's usability. Additionally, participants rated MTMT's effectiveness in identifying and addressing security threats. The open-ended questions were included to gather qualitative insights, prompting students to elaborate on the strengths and weaknesses of MTMT and suggest potential improvements to the tool or the experimental process. This mixed-methods approach ensured a comprehensive understanding of both the tool's functionality and the user experience.

The effectiveness instrument involves threat model submissions based on the accuracy of their DFDs, ensuring that system components, data flows, and trust boundaries are correctly represented and adhere to the case study answer scheme. Additionally, the proposed mitigations for each identified threat were assessed for their relevance and appropriateness in addressing specific vulnerabilities. The overall score for all students provided a holistic view of MTMT's effectiveness, measured by task completion rates and the proportion of tasks with errors, as described in [17].

The usability instrument involves the System Usability Scale (SUS) measurement [18] (which consists of ten questions that

assess various aspects of usability, such as ease of use, complexity, and user confidence interacting with the MTMT tool based on submitted 10 Likert-scale ratings.

D. Phase 4: Data Analysis

In this phase, the collected data were systematically analysed to evaluate the effectiveness and usability of the MTMT among undergraduate students. The tool's effectiveness was measured using the two formulas: Task completion rate and Proportion of tasks with error, as defined in ISO/IEC 25022:2016(E) [17].

Data analysis was performed using SPSS and Microsoft Excel, applying a mix of descriptive statistics and statistical tests. The statistical analysis evaluating the relationship between task complexity and usability rates uses several methods to provide a comprehensive understanding of the data. The following statistical methods were used: quantitative measures, including task completion rates and error rates, were analysed to assess students' efficiency and consistency in completing assigned tasks. The task completion rate is expressed as a percentage, representing the ratio of completed tasks to the total number of tasks. Statistical measures, such as the mean, median, or standard deviation of task completion rates across participants, can be calculated from the produced task completion rates. Errors made during the task were summarised to identify complexity levels and challenging areas in the MTMT.

SUS were calculated to assess the perceived usability of MTMT. The scores were categorised into ranges (e.g., 0–50, 51–68, 69–80, and 80–100) to evaluate the tool's effectiveness in meeting user expectations (Brooke, 1996). This analysis provided insights into the tool's ease of use and alignment with educational objectives. Overall, the data analysis phase combined statistical rigour with qualitative feedback to provide a comprehensive evaluation of MTMT's effectiveness and usability as a teaching and learning tool. This dual approach enabled a deeper understanding of the tool's strengths and areas for improvement.

The pre-experimental setup for the MTMT initiative was carefully designed to equip participants with the necessary knowledge and skills before engaging in practical case studies. Initially, two pre-recorded lecture videos were designed to provide a basic overview of MTMT, emphasising its importance in detecting and mitigating security risks. These lectures not only explained theoretical principles but also offered practical insights into the tool's functionality. To strengthen this theoretical foundation, two hands-on demonstration videos were prepared to guide participants through the key elements of MTMT. These demonstrations covered essential topics, including producing DFDs, proposing security mitigations, generating threat model reports, and applying these skills to a case study.

As shown in Fig. 1, the MTMT video-based learning resources include a comprehensive set of materials designed to enhance understanding and practical application. The resources feature a pre-recorded lecture video introducing MTMT, its application in threat modelling, and foundational concepts. Additionally, two hands-on demo videos provide step-by-step practical demonstrations, guide users through the implementation of core functionalities and the system's main features, and offer guidance on how to solve the task in the case study example. These videos emphasise essential tasks and best practices, ensuring effective learning and obtaining a good understanding of the threat model concept before solving the given case study. Together, these resources provide students with a well-rounded educational experience on the MTMT tool.

Before evaluating the submitted threat models, the files were downloaded and stored on the researcher's laptop in a dedicated folder. Each file was labelled with the student's matric number and name to ensure accurate identification. The folder served as a local repository for all submissions, with strict security measures implemented to ensure confidentiality. Access to the data was restricted to the researcher and the project supervisor, in accordance with ethical standards, to safeguard participants' privacy throughout the study.

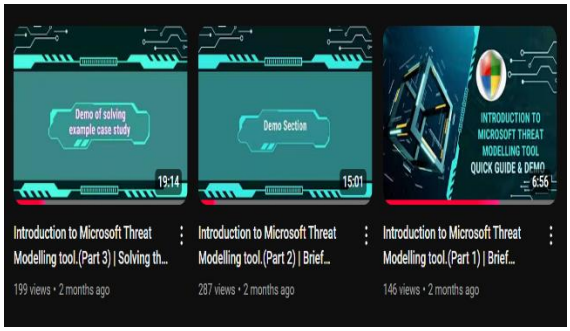

Fig. 1. Three video-based learning resources.

IV. Results

The collected data was analysed using a statistical analysis tool and spreadsheet software. A descriptive analysis was conducted to summarise and comprehensively explore the dataset. Key statistical measures—mean, median, mode, standard deviation, variance, and range — were calculated to provide insights into the data distribution. In this section, the effectiveness of the threat model software will be assessed using two key metrics: task completion rate and the proportion of tasks with errors. The task completion rate will provide a clear indication of how successfully users can complete their intended tasks. In contrast, the proportion of tasks with errors will highlight areas where users encounter difficulties, thereby revealing potential issues with effectiveness.

Additionally, the usability aspect will be evaluated using the SUS, which offers a standardised measure of user satisfaction. By analysing the collected SUS scores and their distribution across various grades, this will provide insights into the overall user experience, identifying strengths and weaknesses in the system's design and functionality. In total, forty-eight undergraduate students participated, submitting their threat model reports and completing the feedback form, which was designed to capture insights into their experience with the tool, usability evaluation, and future adoption likelihood.

A. Effectiveness Aspect

1) Task completion rate: Table I presents the first effectiveness measurement of the MTMT, based on task completion rates. Task 1 achieved significantly higher

performance, with participants collectively scoring 806 out of 1008 marks, resulting in an 80% completion rate. Conversely, task 2 resulted in only 204 out of 384 marks, corresponding to a 53% completion rate. This disparity suggests that Task 1 was either easier to complete or better understood by participants. The template is designed so that author affiliations are not repeated for multiple authors with the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.

Table I presents the first effectiveness measurement of the MTMT, based on task completion rates. Task 1 achieved significantly higher performance, with participants collectively scoring 806 out of 1008 marks, resulting in an 80% completion rate. Conversely, task 2 resulted in only 204 out of 384 marks, corresponding to a 53% completion rate. This disparity suggests that Task 1 was either easier to complete or better understood by participants.

TABLE I. PARTIAL DATASET OF TASK COMPLETION RATE

Student ID	Task 1 (create a DFD) Score /21	Task 2 (mitigation strategies) Score /8	Total Score /29	Total Score %
XXXX	16	1	17	58.62
XXXX	17	3	20	68.97
XXXX	18	7	25	86.2
XXXX	18	7	25	86.21
XXXX	19	0	19	65.51
..	..	..	..	..

Descriptive Statistics

	N	Sum
Task 1 Score	48	806
Task 2 Score	48	204
Valid N (listwise)	48	

Fig. 2. Table of the sum of tasks completed for each task.

As shown in Fig. 2 and Fig. 3, the data reveal that the participants generally performed better on Task 1 than on Task 2, as reflected by higher mean, median, and sum values for Task 1. The sum of all participants' scores for Task 1 is 806, while for Task 2, it totals 204. This indicates higher cumulative performance in Task 1 than in Task 2. The combined total scores suggest an overall moderate performance across both tasks, with a mean of approximately 21 out of a possible 29 points.

The overall task completion rate equation is given in Eq. (1):

$$\text{Task completion rate} = 1010/1392 \times 100 \quad (1)$$

$$= 0.7256 \text{ or } 72.56\%$$

where,

x= Completion rate

a = Number of marks successfully achieved: 06+204=1010

b = Total number of marks: 1392

The overall performance, with a completion percentage of 72.56%, shows that respondents completed a significant portion of the tasks or marks. However, this result indicates that there is still potential for improvement to achieve optimal performance levels closer to 100%. The 382 documented errors had a substantial impact on completion rate, indicating potential issues with usability, task complexity, or respondent understanding. A completion rate below 80% often indicates usability issues, such as interface complexity, insufficient tool support, or task-completion challenges that require further analysis and improvement.

2) Proportion of tasks with errors: Table II presents the second effectiveness measurement of the MTMT, which was assessed as the proportion of tasks with errors. In Task 1, there were 202 errors out of a total of 1008 marks, resulting in an error percentage of approximately 20%, indicating relatively successful task completion with fewer mistakes. Task 2 had 180 errors out of 384 marks, resulting in an error rate of roughly 47%. This suggests that users encountered more challenges when completing Task 2, highlighting areas for potential improvement in the tool to enhance user effectiveness and reduce errors.

TABLE II. PARTIAL DATASET OF THE PROPORTION OF TASKS WITH ERRORS

Student ID	Task 1 (create a DFD) score /21	Task 2 (mitigation strategies) Score /8	Total Error /29	Total Error %
XXXX	5	7	12	41.38
XXXX	4	5	9	31.03
XXXX	3	1	4	13.8
XXXX	3	1	4	13.8
XXXX	2	8	10	34.5
..	..	..	..	..

Statistics

		Task 1 /21	Task 2 /8	Total Error /29
N	Valid	48	48	48
	Missing	0	0	0
Mean		4.21	3.75	7.96
Median		3.00	3.00	7.00
Mode		3	1	4
Std. Deviation		3.108	2.572	4.267
Variance		9.658	6.617	18.211
Range		15	7	19

Fig. 3. The results obtained from the task completion rate dataset.

Proportion of "Tasks with errors" equation is given in Eq. (2). The project conducted a usability test with 48 students, each of whom could earn up to 29 total marks, for a maximum of 1392. Errors were recorded in 382 tasks.

$$\text{Proportion of Task with Errors} = 382/1394 \quad (2)$$

$$= 0.2744 \text{ or } 27.44\ \%$$

where,

x= Proportion of tasks with errors

a = Number of tasks with errors: 382

b = Total marks: 1392

27.44% of the tasks performed by users included at least one error. This metric can be used to assess the usability of the system, where a lower percentage indicates fewer user errors and better usability.

B. Usability Aspect

1) System usability scale (SUS) analysis: The SUS is calculated using a specific equation applied to students' responses to a 10-item questionnaire [18]. Each item is rated on a 5-point Likert scale, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). The following is a standard range of SUS scores and their corresponding:

The steps to calculate the SUS grade

1) Scoring odd-numbered questions (Positive statements): For items 1, 3, 5, 7, and 9, subtract one from the user's response. Score = Response - 1.
2) Scoring even-numbered questions (Negative statements): For items 2, 4, 6, 8, and 10, subtract the user's response from 5. Score = 5 - Response
3) Summing the scores: Sum up all the adjusted scores from steps 1 and 2. This sum is the raw SUS score.
4) Multiply by 2.5: To convert the raw SUS score into a value ranging from 0 to 100, multiply the total by 2.5.

SUS Score = (Sum of Adjusted Scores) × 2.5

For example:

Raw SUS Score = 3 + 3 + 4 + 4 + 2 + 1 + 4 + 3 + 2 + 1 = 27

SUS Score = 27 × 2.5 = 67.5

TABLE III. THE RANGE OF SCORES APPLIED IN THE SUS SCORE GRADES

SUS Score	Grade	Adjective Rating
>80.3	A	Excellent
68-80.3	B	Good
68	C	Okay
51-68	D	Poor
<51	E	Awful

Statistics

SUS Final Score

N	Valid	48
	Missing	0
Mean		59.479
Median		61.250
Mode		67.5
Std. Deviation		14.6770
Variance		215.414
Range		75.0

Fig. 4. The results obtained from the SUS dataset.

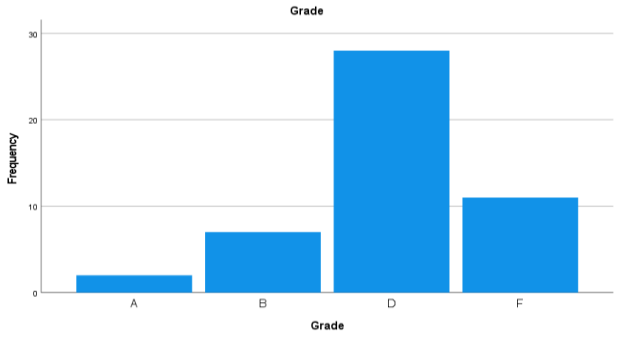

Fig. 5. The frequency table of grade distribution.

TABLE IV. THE FREQUENCY COUNT FROM EACH GRADE DISTRIBUTION

Grade	A	B	C	D	F
Count	2	7	0	28	11

TABLE V. FREQUENCY COUNT AND PERCENTAGE FROM EACH PROFICIENCY LEVEL

Proficiency Level	Count	Percentage (%)
Advanced (Grade A & B) (69%-100%)	9	18.75
Intermediate (Grade D) (51-68)	28	58.33
Basic (Grade E) (0-50)	11	22.92

V. DISCUSSION

The obtained results, in terms of effectiveness, highlight both strengths and areas for improvement in the MTMT. Task 1 demonstrated a significantly higher completion rate of 80% compared to Task 2's 53%, suggesting that participants either found Task 1 more straightforward or better understood it. However, the overall completion rate of 72.56% indicates that, while a significant number of tasks were completed, there is still room to grow to achieve ideal performance around 100%. Error analysis further underscores this need: Task 1 had a relatively low error rate of 20%, whereas Task 2 had a higher rate of 47%. This gap shows that users had more difficulty with Task 2, potentially due to task complexity, usability issues, or insufficient tool support for entering the proposed mitigation strategies for each identified threat. The combined total of 382 errors had a significant impact on the overall completion rate, highlighting the importance of fixing these issues to improve user effectiveness. These errors, which accounted for 27.44% of the tasks users performed, underscore areas where the system's usability can be improved. Addressing these issues would not only reduce the error rate but also likely lead to higher task success rates and better user confidence. By focusing on error-prone areas, the system can improve efficiency, ensuring a smoother, more intuitive user experience. This aligns with the usability goals outlined in [17], emphasising the need for continuous improvement to meet user needs effectively. These findings highlight the importance of improving the tool's interface and support features to minimise errors and increase task completion rates across all task types.

The overall usability rating indicates that 59.5 is below average, as it falls short of the industry-standard benchmark of

68. This suggests moderate usability challenges with the MTMT. The frequency table and bar chart of grade distribution, as shown in Fig. 4, Fig. 5, Table III and Table IV, reveal that many students received a grade 'D', with a total count of 28. This represents the highest frequency among all grades, indicating a significant portion of the class struggled to achieve higher performance levels. Grade 'F' follows with a count of 11, further emphasising the challenges students face in meeting the required standards. This distribution demonstrates an overall pattern toward lower grades. Based on Table V, the usability assessment also highlights that 58.3% of users rated the MTMT as moderately challenging, which can be attributed to their lack of experience with threat modelling tools. As first-time users, the students faced a learning curve in understanding the tool's interface, functionalities, and application within the context of threat modelling. This indicates that they require more time to study and understand the process of creating data flow diagrams and evaluating the tool's automated threat detection, as they lacked a basic understanding of another threat modelling tool. This lack of prior exposure indeed led them to rate the tool as moderately challenging, even though it is designed to simplify threat modelling for users with varying cybersecurity skills.

The evaluation of the MTMT reveals important insights regarding its effectiveness and usability. The overall task completion rate of 72.56% indicates that while a substantial portion of tasks were completed, there remains significant potential for improvement to reach optimal performance levels. The identification of 382 documented errors highlights usability issues, task complexity, and potential user misunderstandings. With 27.44% of tasks containing at least one error, this metric underscores the need for further analysis to enhance usability. The mean SUS score of 59.5, which falls below the industry benchmark of 68, suggests moderate usability challenges that affect user experience. The considerable standard deviation of 14.67 indicates significant variability in user perceptions, with scores ranging from 25 to 100 reflecting extreme differences in usability experiences. Notably, only 18.6% of users rated the tool as highly usable, while 58.3% encountered moderate challenges, indicating a clear need for improvements in design and support.

These findings emphasise the need for targeted enhancements to the tool's interface and functionality to better support users and improve overall effectiveness in educational contexts. While students completed a substantial portion of the assigned tasks, there remains considerable room for improvement, with 382 documented errors impacting performance. The System Usability Scale assessment yielded a mean score of 59.5, which falls below the industry-standard benchmark of 68. The varying user experiences are reflected in the score distribution: only 18.6% of users found the tool highly usable (Grades A and B combined), while 58.3% encountered moderate challenges.

VI. Conclusion

The study employed a structured methodology comprising four phases: content analysis, experiment design, data collection, and data analysis. In the initial phase, learning materials and an online form were developed to guide participants through the experiment. During the experiment design phase, students were introduced to the MTMT using the prepared learning materials. Data on usability and effectiveness were collected via online forms in phase three, followed by statistical analysis using spreadsheet software in the final phase. The study found that Task 1 had a higher completion rate than Task 2, with an overall success rate of 72.56%. Task 2 had more errors impacting effectiveness. The SUS score indicates moderate usability issues with the MTMT. Most students found the tool challenging and received low grades, mainly because of limited prior experience.

These results highlight the need to improve MTMT's interface and support features, particularly by allowing users to propose mitigation strategies within MTMT to reduce errors and enhance task completion rates. Addressing these issues would improve user satisfaction and better align the tool with usability standards.

The study’s novelty stems from its dual-metric evaluation: it measured the tool’s effectiveness using the ISO/IEC 25022:2016(E) effectiveness formula and assessed its usability through the System Usability Scale (SUS) analysis, providing a standardised, quantitative assessment framework.

Future work may include a comprehensive comparison of MTMT's usability and feature set with those of alternative threat modelling tools to benchmark its performance, adaptability, and user experience across different contexts. Enhancements such as automated mitigation recommendations could be explored to improve MTMT’s functionality, especially for novice users in educational or training environments, thereby increasing its pedagogical value and practical relevance. The study also integrates real-world cybersecurity tasks using MTMT with learning outcomes on applied threat modelling, ensuring assessments accurately measure the same experiential competencies students develop through hands-on, practice-based learning activities. The data analysis can also be improved by using inferential statistics such as the T-test, ANOVA and correlation.

References

[1] Microsoft, "Getting started - Microsoft Threat Modeling Tool - Azure," Microsoft Learn, 2022. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-getting-started.

[2] Oladimeji, E. A., Supakkul, S., & Chung, L. (2006). Security threat modelling and analysis: A goal-oriented approach. Proceedings of the 10th IASTED International Conference on Software Engineering and Applications, 13–15.

[3] McGraw, G. (2006). Software security: Building security in. Addison-Wesley Professional.

[4] Scandariato, R., Wuyts, K., & Joosen, W. (2013). A descriptive study of Microsoft's threat modelling technique. Requirements Engineering, 18(1), 51–81. https://doi.org/10.1007/s00766-012-0163-2.

[5] Myagmar, S., Lee, A. J., & Yurcik, W. (2005). Threat modelling as a basis for security requirements. Symposium on Requirements Engineering for Information Security, 1–8.

[6] Schneier, B. (2000). Secrets and lies: Digital security in a networked world. John Wiley & Sons.

[7] Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). The EigenTrust algorithm for reputation management in P2P networks. Proceedings of the 12th International Conference on World Wide Web, 640–651.

[8] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Uncover security design flaws using the STRIDE approach," MSDN Magazine, Nov. 2006.

[9] P. Torr, "Demystifying the threat-modelling process," IEEE Security & Privacy, vol. 3, no. 5, pp. 66–70, 2005.

[10] R. Scandariato, K. Wuyts, and W. Joosen, "A descriptive study of Microsoft's threat modelling technique," Requirements Engineering, vol. 18, no. 1, pp. 51–81, 2013. [Online]. Available: https://doi.org/10.1007/s00766-012-0163-2.

[11] M. Howard and S. Lipner, The Security Development Lifecycle: SDL, a Process for Developing Demonstrably More Secure Software. Redmond, WA, USA: Microsoft Press, 2006.

[12] I. Williams and X. Yuan, "Evaluating the effectiveness of the Microsoft threat modelling tool," in Proc. 2015 Information Security Curriculum Development Conf., 2015.

[13] R. Khan, E. Da Silva, T. Pommier, and S. Bouzefrane, "Automated ICS template for STRIDE Microsoft Threat Modeling Tool," in Proc. ARES 2023 Conf., 2023.

[14] L. Verheyden, Effectiveness of Threat Modelling, Master's thesis, Ghent University, 2018. [Online]. Available: https://lib.ugent.be/fulltxt/RUG01/002/508/960/RUG01-002508960_2018_0001_AC.pdf.

[15] J. Selin, Evaluation of Threat Modelling Methodologies: A Case Study, Bachelor's thesis, School of Technology, Information and Communication Technology, 2019. [Online]. Available: http://urn.fi/URN:NBN:fi:amk-2019060615264.

[16] K. Graffi, Z. Shi, D. Starobinski, and N. Matyunin, "Threat modeling tools: A taxonomy," IEEE Security & Privacy, vol. 20, no. 4, pp. 29–39, 2022. [Online]. Available: https://doi.org/10.1109/MSEC.2021.3125229.

[17] International Organization for Standardization, & International Electrotechnical Commission. (2016). Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – Measurement of quality in use (ISO/IEC 25022:2016(E)). First edition, published June 15, 2016. ISO/IEC.

[18] J. Brooke, "SUS: A quick and dirty usability scale," in Usability Evaluation in Industry, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194.

[19] K. Graffi, Z. Shi, D. Starobinski, and N. Matyunin, "Threat modeling tools: A taxonomy," IEEE Security & Privacy, vol. 20, no. 4, pp. 29–39, 2022, doi: 10.1109/MSEC.2021.3125229.